LilyPond

El tipografiador de música

Manual de referencia de la notación

El equipo de desarrollo de LilyPond

Copyright © 1999–2008 por los autores

La traducción de la siguiente nota de copyright se ofrece como cortesía para las personas de habla no inglesa, pero únicamente la nota en inglés tiene validez legal.

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

> Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, versión 1.1 o cualquier versión posterior publicada por la Free Software Foundation; sin ninguna de las secciones invariantes. Se incluye una copia de esta licencia dentro de la sección titulada "Licencia de Documentación Libre de GNU".
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled "GNU Free Documentation License".

Para LilyPond versión 2.12.1

Índice General

1 Notación musical 1

1.1 Alturas 1
1.1.1 Escritura de notas 1
Escritura de octava absoluta 1
Escritura de octava relativa 2
Alteraciones accidentales 4
Nombres de las notas en otros idiomas 6
1.1.2 Modificación de varias notas a la vez 7
Comprobación de octava 8
Transposición 9
1.1.3 Imprimir las alturas 12
Clave 12
Armadura de la tonalidad 15
Corchetes de octava 16
Transposición de los instrumentos 17
Alteraciones accidentales automáticas 19
Tesitura 25
1.1.4 Cabeza de las notas 27
Cabezas de nota especiales 27
Cabezas de notas de Notación Fácil 28
Cabezas de notas con formas diversas 28
Improvisación 30
1.2 Duraciones 31
1.2.1 Escritura de las duraciones (valores rítmicos) 31
Duraciones 31
Grupos especiales 33
Escalar las duraciones 35
Ligaduras de unión 36
1.2.2 Escritura de silencios 39
Silencios 39
Silencios invisibles 41
Silencios de compás completo 42
1.2.3 Impresión de las duraciones 46
Indicación de compás 46
Anacrusas 48
Música sin compasear 49
Notación polimétrica 50
División automática de las notas 53
Mostrar los ritmos de la melodía 53
1.2.4 Barras 56
Barras automáticas 56
Establecer el comportamiento de las barras automáticas 58
Barras manuales 67
Barras progresivas 69
1.2.5 Barras de compás 70
Barras de compás 70
Numeración de compases 73
Comprobación de compás y de número de compás 76

Llamadas de ensayo . . . 77
1.2.6 Asuntos rítmicos especiales . . . 78
Notas de adorno . . . 78
Alinear con una cadenza . . . 82
Gestión del tiempo . . . 83
1.3 Expresiones . . . 84
1.3.1 Adosado a las notas . . . 84
Articulaciones y ornamentos . . . 84
Matices dinámicos . . . 86
Indicaciones dinámicas contemporáneas . . . 90
1.3.2 Curvas . . . 92
Ligaduras de expresión . . . 92
Ligaduras de fraseo . . . 93
Marcas de respiración . . . 94
Caídas y elevaciones . . . 95
1.3.3 Líneas . . . 96
Glissando . . . 96
Arpegio . . . 97
Trinos . . . 100
1.4 Repeticiones . . . 101
1.4.1 Repeticiones largas . . . 102
Repeticiones normales . . . 102
Marcas de repetición manual . . . 105
Repeticiones explícitas . . . 107
1.4.2 Repeticiones cortas . . . 108
Repeticiones de compás o parte de ellos . . . 108
Repeticiones de trémolo . . . 110
1.5 Notas simultáneas . . . 111
1.5.1 Una voz única . . . 111
Notas en acorde . . . 111
Expresiones simultáneas . . . 112
Racimos (clusters) . . . 112
1.5.2 Varias voces . . . 113
Polifonía en un solo pentagrama . . . 113
Estilos de voz . . . 116
Resolución de las colisiones . . . 116
Combinación automática de las partes . . . 120
Escribir música en paralelo . . . 123
1.6 Notación de los pentagramas . . . 126
1.6.1 Impresión de los pentagramas . . . 126
Crear instancias de pentagramas nuevos . . . 126
Agrupar pentagramas . . . 127
Grupos de pentagramas anidados . . . 131
1.6.2 Modificación de pentagramas sueltos . . . 133
El símbolo del pentagrama . . . 133
Pentagramas de Ossia . . . 135
Ocultar pentagramas . . . 139
1.6.3 Escritura de las particellas . . . 142
Indicaciones metronómicas . . . 142
Nombres de instrumentos . . . 145
Citar otras voces . . . 148
Formateo de las notas de aviso . . . 151
1.7 Anotaciones editoriales . . . 154
1.7.1 Dentro del pentagrama . . . 154

Seleccionar el tamaño de la tipografía para la notación . . . 154
Indicaciones de digitación . . . 155
Notas ocultas . . . 158
Colorear objetos . . . 158
Paréntesis . . . 160
Plicas . . . 160
1.7.2 Fuera del pentagrama . . . 161
Globos de ayuda . . . 161
Líneas de rejilla . . . 162
Corchetes de análisis . . . 164
1.8 Texto . . . 165
1.8.1 Escritura del texto . . . 166
Guiones de texto . . . 166
Extensiones de texto . . . 167
Indicaciones de texto . . . 167
Texto separado . . . 171
1.8.2 Formatear el texto . . . 173
Introducción al marcado de texto . . . 173
Instrucciones de marcado de texto más usuales . . . 175
Alineación de texto . . . 177
Notación gráfica dentro de elementos de marcado . . . 180
Notación musical dentro de elementos de marcado . . . 183
Elementos de marcado de varias páginas . . . 185
1.8.3 Tipografías . . . 186
Explicación de las fuentes tipográficas . . . 186
Fuentes de un solo elemento . . . 188
Fuentes tipográficas del documento completo . . . 188

2 Notación especializada . . . 190
2.1 Música vocal . . . 190
2.1.1 Notación común para música vocal . . . 190
Referencias para música vocal y letra . . . 190
Ópera . . . 190
Cancioneros . . . 190
Música hablada . . . 191
Cantos . . . 191
Música vocal antigua . . . 191
2.1.2 Introducir la letra . . . 191
Explicación de la letra . . . 191
Elaborar canciones sencillas . . . 193
Trabajar con el texto y las variables . . . 194
2.1.3 Alineación de la letra a una melodía . . . 194
Duración automática de las sílabas . . . 195
Duración manual de las sílabas . . . 196
Varias sílabas sobre una nota . . . 196
Varias notas sobre una sílaba . . . 197
Saltar notas . . . 198
Líneas de extensión y guiones . . . 198
Letras y repeticiones . . . 199
2.1.4 Usos específicos de la letra . . . 199
Letras en divisi . . . 199
Letra independiente de las notas . . . 200
Espaciado de las sílabas . . . 201
Centrado de la letra entre pentagramas . . . 202

2.1.5 Versos ... 202
Añadir números de verso ... 202
Añadir expresiones dinámicas a los versos ... 202
Añadir el nombre de los cantantes a los versos ... 203
Versos con ritmos distintos ... 203
Imprimir los versos al final ... 205
Imprimir los versos al final en varias columnas ... 206
2.2 Teclados y otros instrumentos de varios pentagramas ... 207
2.2.1 Notación común para instrumentos de teclado ... 208
Referencias para teclados ... 208
Cambiar de pentagrama manualmente ... 209
Cambiar de pentagrama automáticamente ... 210
Líneas de cambio de pentagrama ... 211
Plicas de pentagrama cruzado ... 212
2.2.2 Piano ... 214
Pedales de piano ... 214
2.2.3 Acordeón ... 215
Símbolos de Discanto ... 215
2.2.4 Arpa ... 218
Referencias para notación de arpa ... 218
Pedales de arpa ... 219
2.3 Instrumentos de cuerda sin trastes ... 219
2.3.1 Notación común para cuerdas sin trastes ... 220
Referencias para cuerdas sin trastes ... 220
Indicaciones de arco ... 220
Armónicos ... 221
Snap (Bartók) pizzicato ... 222
2.4 Instrumentos de cuerda con trastes ... 223
2.4.1 Notación común para cuerdas con trastes ... 223
Referencias para cuerdas con trastes ... 223
Indicación de los números de cuerda ... 224
Tablaturas predeterminadas ... 225
Tablaturas personalizadas ... 228
Marcas de diagramas de trastes ... 229
Diagramas predefinidos de trastes ... 238
Diagramas de traste automáticos ... 245
Digitaciones de la mano derecha ... 248
2.4.2 Guitarra ... 249
Indicar la posición y la cejilla ... 250
Indicar armónicos y notas tapadas ... 250
2.4.3 Banjo ... 250
Tablaturas de banjo ... 250
2.5 Percusión ... 251
2.5.1 Notación común para percusión ... 251
Referencias para percusión ... 251
Notación básica de percusión ... 251
Redobles ... 252
Percusión afinada ... 253
Pautas de percusión ... 253
Pautas de percusión personalizadas ... 255
Notas fantasma ... 259
2.6 Instrumentos de viento ... 260
2.6.1 Notación común para instrumentos de viento ... 260
Referencias para instrumentos de viento ... 260

Digitaciones . . . 262
2.6.2 Gaita . . . 262
Definiciones para la gaita . . . 262
Ejemplo de música de gaita . . . 262
2.7 Notación de acordes . . . 264
2.7.1 Modo de acordes . . . 264
Panorámica del modo de acordes . . . 264
Acordes más usuales . . . 265
Acordes extendidos y alterados . . . 266
2.7.2 Imprimir los acordes . . . 269
Impresión de los nombres de acorde . . . 269
Nombres de acorde personalizados . . . 272
2.7.3 Bajo cifrado . . . 276
Introducción al bajo cifrado . . . 276
Introducir el bajo cifrado . . . 277
Imprimir el bajo cifrado . . . 280
2.8 Notación antigua . . . 283
2.8.1 Panorámica de los estilos contemplados . . . 284
2.8.2 Notación antigua: funcionalidades comunes . . . 284
Contextos predefinidos . . . 284
Ligaduras . . . 285
Custos . . . 285
Soporte para bajo cifrado . . . 286
2.8.3 Tipografiar música mensural . . . 286
Contextos de la música mensural . . . 286
Claves de la música mensural . . . 287
Indicaciones de compás de la música mensural . . . 288
Cabezas de nota de la música mensural . . . 289
Corchetes de la música mensural . . . 290
Silencios de la música mensural . . . 291
Alteraciones y armaduras de la música mensural . . . 291
Alteraciones de anotación (*musica ficta*) . . . 292
Ligaduras mensurales blancas . . . 292
2.8.4 Tipografiado del canto gregoriano . . . 294
Contextos del canto gregoriano . . . 294
Claves de canto gregoriano . . . 295
Alteraciones y armaduras de canto gregoriano . . . 296
Divisiones . . . 296
Articulaciones del canto gregoriano . . . 297
Puntos de aumentación (*morae*) . . . 297
Ligaduras de neumas cuadrados gregorianos . . . 298
2.8.5 Trabajar con música antigua: escenarios y soluciones . . . 305
Incipits . . . 305
Disposición tipo «Mensurstriche» . . . 305
Transcripción de canto gregoriano . . . 306
Antigua y moderna desde la misma fuente . . . 309
Anotaciones editoriales . . . 309
2.9 Músicas del mundo . . . 309
2.9.1 Música árabe . . . 309
Referencias para música árabe . . . 309
Nombres de nota árabes . . . 310
Armaduras de tonalidad árabes . . . 311
Indicaciones de compás árabes . . . 312
Ejemplo de música árabe . . . 314

Lecturas adicionales 314

3 Entrada y salida generales 316

3.1 Estructura del código de entrada 316
3.1.1 Estructura de una partitura 316
3.1.2 Varias partituras en un libro 317
3.1.3 Estructura del archivo 318
3.2 Títulos y cabeceras 320
3.2.1 Crear títulos 320
3.2.2 Títulos personalizados 323
3.2.3 Referencias a números de página 324
3.2.4 Índice general 325
3.3 Trabajar sobre los archivos de entrada 327
3.3.1 Inclusión de archivos de LilyPond 327
3.3.2 Distintas ediciones a partir de una misma fuente 328
Uso de las variables 328
Uso de etiquetas 330
3.3.3 Codificación del texto 332
3.3.4 Presentar la notación de LilyPond 333
3.4 Controlar la salida 334
3.4.1 Extracción de fragmentos de música 334
3.4.2 Saltar la música corregida 334
3.5 Salida MIDI 335
3.5.1 Crear archivos MIDI 335
Nombres de instrumentos 335
3.5.2 El bloque MIDI 337
3.5.3 ¿Qué sale por el MIDI? 338
Contemplado en el MIDI 338
No contemplado en el MIDI 338
3.5.4 Repeticiones y MIDI 338
3.5.5 Control de los matices en el MIDI 339
Indicaciones dinámicas 339
Volumen maestro MIDI 340
Balance entre instrumentos (i) 341
Balance entre instrumentos (i) 342
3.5.6 Percusión en MIDI 343

4 Problemas de espaciado 345

4.1 El papel y las páginas 345
4.1.1 Tamaño del papel 345
4.1.2 Formateo de las páginas 346
Dimensiones verticales 346
Dimensiones horizontales 347
Otras variables de diseño de página 348
4.2 Disposición de la música 351
4.2.1 Establecer el tamaño del pentagrama 351
4.2.2 Disposición de la partitura 352
4.3 Saltos 352
4.3.1 Saltos de línea 352
4.3.2 Saltos de página 354
4.3.3 Saltos de página óptimos 355
4.3.4 Paso de página óptimo 355
4.3.5 Saltos de página mínimos 356

4.3.6 Saltos de línea explícitos ... 357
4.3.7 Utilizar una voz adicional para los saltos de línea ... 358
4.4 Espaciado vertical ... 360
4.4.1 Espaciado vertical dentro de un sistema ... 360
4.4.2 Espaciado vertical entre sistemas ... 362
4.4.3 Posicionamiento explícito de los pentagramas y los sistemas ... 365
4.4.4 Espaciado vertical en dos fases ... 371
4.4.5 Evitar las colisiones verticales ... 372
4.5 Espaciado horizontal ... 373
4.5.1 Panorámica del espaciado horizontal ... 373
4.5.2 Área de espaciado nueva ... 375
4.5.3 Cambiar el espaciado horizontal ... 375
4.5.4 Longitud de la línea ... 377
4.5.5 Notación proporcional ... 378
4.6 Encajar la música en menos páginas ... 384
4.6.1 Mostrar el espaciado ... 385
4.6.2 Cambiar el espaciado ... 386

5 Cambiar los valores por omisión ... 388
5.1 Contextos de interpretación ... 388
5.1.1 Explicación de los contextos ... 388
Score: el contexto maestro ... 388
Contextos del nivel superior: contenedores de pentagramas ... 388
Contextos de nivel intermedio: pentagramas ... 389
Contextos del nivel más bajo: voces ... 389
5.1.2 Crear contextos ... 390
5.1.3 Modificar los complementos (plug-ins) de contexto ... 391
5.1.4 Cambiar los valores por omisión de los contextos ... 393
5.1.5 Definir contextos nuevos ... 394
5.1.6 Alinear los contextos ... 395
5.2 Explicación del Manual de referencia de funcionamiento interno ... 397
5.2.1 Navegar por la referencia del programa ... 397
5.2.2 Interfaces de la presentación ... 398
5.2.3 Determinar la propiedad del grob ... 399
5.2.4 Convenciones de nombres ... 400
5.3 Modificar las propiedades ... 400
5.3.1 Panorámica de la modificación de las propiedades ... 400
5.3.2 La instrucción `\set` ... 401
5.3.3 La instrucción `\override` ... 403
5.3.4 La instrucción `\tweak` ... 403
5.3.5 `\set` frente a `\override` ... 405
5.4 Conceptos y propiedades útiles ... 406
5.4.1 Modos de entrada ... 406
5.4.2 Dirección y posición ... 407
5.4.3 Distancias y medidas ... 408
5.4.4 Propiedades del símbolo del pentagrama ... 409
5.4.5 Objetos de extensión ... 410
Uso del `spanner-interface` ... 410
Uso del `line-spanner-interface` ... 412
5.4.6 Visibilidad de los objetos ... 415
Eliminar el sello ... 415
Hacer transparentes los objetos ... 415
Pintar los objetos de blanco ... 415
Uso de break-visibility (visibilidad en el salto) ... 416

Consideraciones especiales 418
5.4.7 Estilos de línea 419
5.4.8 Rotación de objetos 420
Rotación de objetos de presentación 420
Rotación de elementos de marcado 421
5.5 Trucos avanzados 421
5.5.1 Alineación de objetos 421
Establecer directamente `X-offset` y `Y-offset` 422
Uso del `side-position-interface` 422
Uso del `self-alignment-interface` 423
Uso del `break-alignable-interface` 424
5.5.2 Agrupación vertical de objetos gráficos 426
5.5.3 Modificación de los sellos 426
5.5.4 Modificación de las formas 427
Modificación de ligaduras de unión y de expresión 427

6 Interfaces para programadores 429
6.1 Funciones musicales 429
6.1.1 Panorámica de las funciones musicales 429
6.1.2 Funciones de sustitución sencillas 429
6.1.3 Funciones de sustitutión en parejas 431
6.1.4 Matemáticas dentro de las funciones 431
6.1.5 Funciones vacías 432
6.1.6 Funciones sin argumentos 432
6.1.7 Pranorámica de las funciones musicales disponibles 432
6.2 Interfaces para el programador 436
6.2.1 Variables de entrada y Scheme 436
6.2.2 Representación interna de la música 437
6.3 Construcción de funciones complejas 438
6.3.1 Presentación de expresiones musicales 438
6.3.2 Propiedades de la música 439
6.3.3 Doblar una nota con ligaduras (ejemplo) 439
6.3.4 Añadir articulación a las notas (ejemplo) 441
6.4 Interfaz de marcado para el programador 443
6.4.1 Construcción del marcado en Scheme 443
6.4.2 Cómo funciona internamente el marcado 444
6.4.3 Definición de una instrucción de marcado nueva 444
6.4.4 Definición de nuevas instrucciones de lista de marcado 446
6.5 Contextos para programadores 447
6.5.1 Evaluación de contextos 447
6.5.2 Ejecutar una función sobre todos los objetos de la presentación 447
6.6 Procedimientos de Scheme como propiedades 448
6.7 Usar código de Scheme en lugar de `\tweak` 449
6.8 Trucos difíciles 449

Apéndice A Lista bibliográfica 451

Apéndice B Tablas del manual sobre notación 452
B.1 Carta de nombres de acordes 452
B.2 Modificadores de acorde más usuales 453
B.3 Diagramas predefinidos de trastes 456
B.4 Instrumentos MIDI 459
B.5 Lista de colores 460
B.6 La tipografía Feta 461
B.7 Estilos de cabezas de nota 475
B.8 Text markup commands 476
B.8.1 Font 476
B.8.2 Align 485
B.8.3 Graphic 498
B.8.4 Music 502
B.8.5 Instrument Specific Markup 505
B.8.6 Other 508
B.9 Text markup list commands 512
B.10 Lista de articulaciones 513
B.11 Notas de percusión 514
B.12 Todas las propiedades de contexto 515
B.13 Propiedades de disposición 525
B.14 Identificadores 538
B.15 Funciones de Scheme 542

Apéndice C Hoja de referencia rápida 562

Apéndice D Licencia de documentación libre de GNU 566
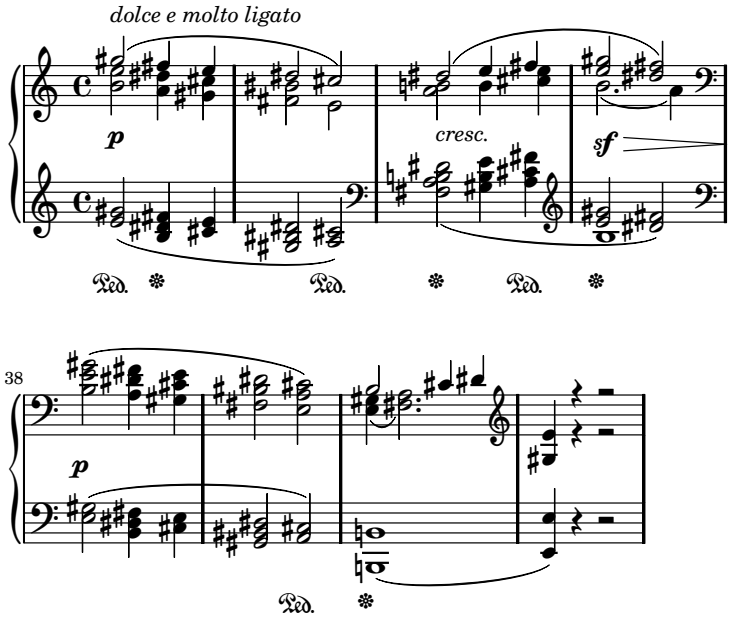
Apéndice E Índice de instrucciones de LilyPond 572

Apéndice F Índice de LilyPond 579

1 Notación musical

Este capítulo explica cómo crear notación musical.

1.1 Alturas



En esta sección se discute cómo especificar la altura de las notas. Este proceso se compone de tres fases: entrada, modificación y salida.
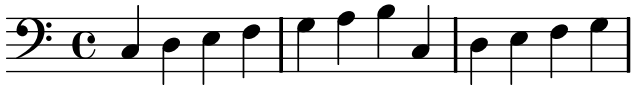
1.1.1 Escritura de notas

En esta sección se describe la manera de introducir la altura de las notas. Esisten dos formas distintas de colocar las notas en su octava correspondiente: el modo absoluto y el relativo. En casi todas las ocasiones, será más práctico el modo relativo.

Escritura de octava absoluta

El nombre de una nota se especifica usando las letras minúsculas de la `a` a la `g`. Las notas cuyos nombres van desde `c` hasta `b` se imprimen en la octava inferior al Do central.
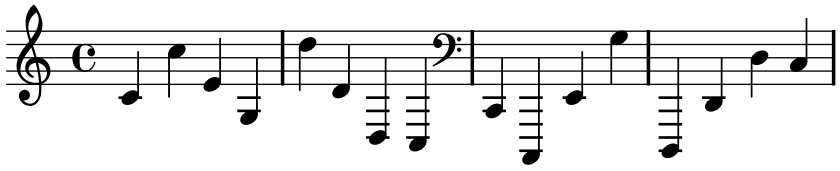
```
\clef bass
c d e f
g a b c
d e f g
```



Se pueden especificar otras octavas mediante una comilla simple quote (`'`) o una coma (`,`) . Cada `'` eleva la altura en una octava; cada `,` baja la altura una octava.

```
\clef treble
c' c'' e' g
d'' d' d c
\clef bass
```

```
c, c,, e, g
d,, d, d c
```



Véase también

Glosario musical: Sección “Nombres de las notas” in *Glosario Musical*.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Escritura de octava relativa

Cuando las octavas se especifican en modo absoluto, es fácil poner por accidente una nota en la octava equivocada. El modo de octavas relativas reduce estos errores porque casi todo el tiempo es innecesario indicar octava alguna. Es más, en el modo absoluto puede ser difícil encontrar un error aislado, mientras que en el modo relativo un solo fallo hace que el resto de la pieza esté una octava más alta o más baja de lo que debería.
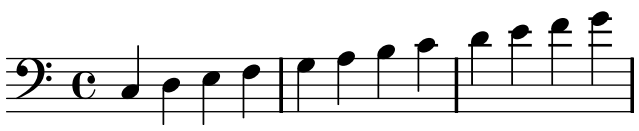
`\relative` *`altura_inicial expresión_musical`*

En el modo relativo, se supone que cada nota se encuentra lo más cerca posible de la nota anterior. Esto significa que la octava de una nota que está dentro de *expresión_musical* se calcula como sigue:

- Si no se usa ninguna marca de cambio de octava en una nota, su octava se calcula de forma que el intervalo que forme con la nota anterior sea menor de una quinta. Este intervalo se determina sin considerar las alteraciones.
- Se puede añadir una marca de cambio de octava `'` o `,` para elevar o bajar la altura, respectivamente, en una octava más en relación con la altura calculada sin esta marca.
- Se pueden usar varias marcas de cambio de octava. Por ejemplo, `''` y `,,` alteran la altura en dos octavas.
- La altura de la primera nota es relativa a *`altura_inicial`*. *altura_inicial* se especifica en modo de octava absoluta, y se recomienda que sea un Do (una octava de `c`).

Aquí podemos ver el modo relativo en acción:
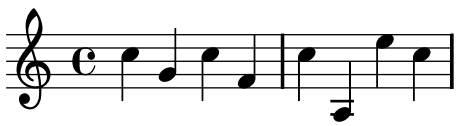
```
\relative c {
  \clef bass
  c d e f
  g a b c
  d e f g
}
```



Las marcas de cambio de octava se utilizan para intervalos mayores de la cuarta:
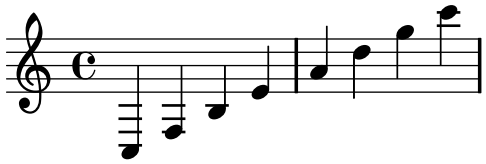
```
\relative c'' {
  c g c f,
  c' a, e'' c
```
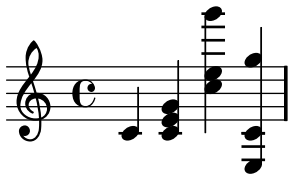
```
}
```



Una serie de notas sin ninguna marca de octava puede, a pesar de todo, abarcar intervalos muy grandes:

```
\relative c {
  c f b e
  a d g c
}
```
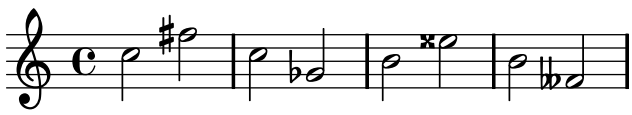


Si el elemento anterior es un acorde, la primera nota del acorde se utiliza para determinar la primera nota del siguiente acorde. Dentro de los acordes, la siguiente nota siempre está en relación a la anterior.

```
\relative c' {
  c
  <c e g>
  <c' e g'>
  <c, e, g''>
}
```



Como se explicó más arriba, la octava de las notas se calcula solamente a partir de sus nombres, sin tener encuenta alteración alguna. Por tanto, un Mi doble sostenido después de un Si se escribirá más agudo, mientras que un Fa doble sostenido se escribirá más grave. En otras palabras, se considera a la cuarta doble aumentada un intervalo menor que la quinta doble disminuida, independientemente del número de semitonos de cada uno de ellos.

```
\relative c'' {
  c2 fis
  c2 ges
  b2 eisis
  b2 feses
}
```



Véase también

Glosario musical: Sección “quinta” in *Glosario Musical*, Sección “intervalo” in *Glosario Musical*, Sección “Nombres de las notas” in *Glosario Musical*.

Referencia de la notación: [Comprobación de octava], página 8.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RelativeOctaveMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La conversión relativa no afecta a las secciones `\transpose`, `\chordmode` o `\relative` en su argumento. Para usar el modo relativo dentro de música transportada, se debe escribir un `\relative` adicional dentro de `\transpose`.
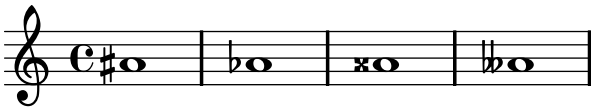
Si no se especifica ninguna *altura_inicial* para `\relative`, entonces se supone que es `c'`. Sin embargo, ésta es una opción en desuso y podría desaparecer en versiones posteriores, por lo que su utilización está desaconsejada.

Alteraciones accidentales

> **Nota:** Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios. En LilyPond, los nombres de las notas son el código de entrada en bruto; la armadura y la clave determinan de qué forma se presenta este código en bruto. Una nota sin alteración como `c` significa ‘Do natural’, si que le afecten la armadura ni la clave. Para ver más información, consulte Sección “Accidentals and key signatures” in *Manual de Aprendizaje*.
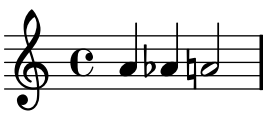
Se escribe una nota *sostenida* añadiendo `is` al nombre de la nota, y un *bemol* añadiendo `es`. Como es de esperar, un *dobles sostenido* y un *doble bemol* se obtiene añadiendo `isis` o `eses`. Esta sintaxis deriva de los nombres de las notas en holandés. Para utilizar otros nombres para las alteraciones, consulte [Nombres de las notas en otros idiomas], página 6.

```
ais1 aes aisis aeses
```



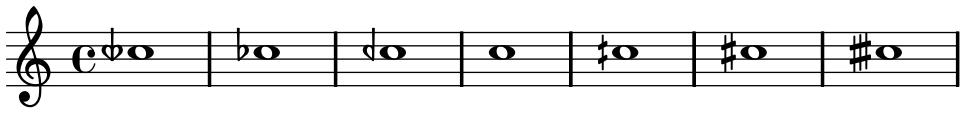
Un becuadro cancelará el efecto de una alteración accidental o de la armadura. Sin embargo, los becuadros no se codifican dentro de la sintaxis del nombre de la nota con un sufijo; una nota becuadro se muestra como un simple nombre de nota natural:

```
a4 aes a2
```
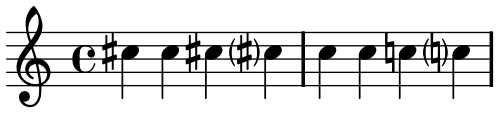


Se pueden escribir medios bemoles y los medios sostenidos; a continuación presentamos una serie de DOs cada vez más agudos:

```
ceseh1 ces ceh c cih cis cisih
```



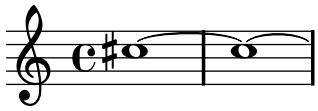
Normalmente las alteraciones accidentales se imprimen automáticamente, pero también puede imprimirlas manualmente. Un alteración recordatoria se puede forzar añadiendo un signo de admiración `!` después de la altura de la nota. Se puede obtener una alteración de precaución (o sea, una alteración entre paréntesis) añadiendo el signo de interrogación `?` después del nombre de la nota. Estas alteraciones adicionales se pueden usar también para producir notas con becuadro.
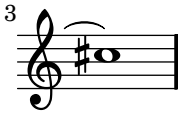
```
cis cis cis! cis? c c c! c?
```



Las alteraciones sobre notas unidas por ligadura sólo se imprimen al comienzo de un sistema:

```
cis1 ~ cis ~
\break
cis
```
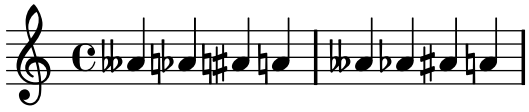




Fragmentos de código seleccionados

Evitar que se añadan becuadros adicionales automáticamente

Según las reglas estándar de composición tipográfica, se imprime un becuadro antes de un sostenido o un bemol cuando se tiene que cancelar una alteración anterior en la misma nota. Para modificar este comportamiento, establezca el valor de la propiedad `extraNatural` a `##f` (falso) dentro del contexto de `Staff`.

```
\relative c'' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```
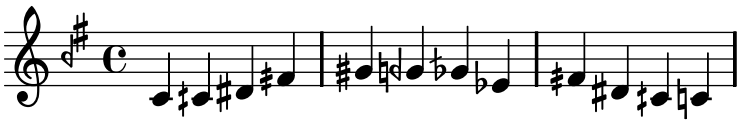


El «Makam»

El «Makam» es un tipo de melodía de Turquía que utiliza intervalos microtonales de 1/9 de tono. Consulte el archivo de inicio `makam.ly` (véase el 'Manual de aprendizaje 2.12.1, 4.6.3 Otras fuentes de información' para averiguar la situación de este archivo) para ver detalles de los nombres de las notas y las alteraciones.

```
% Iniciar ajustes de makam
\include "makam.ly"

\relative c' {
  \set Staff.keySignature = #`((3 . ,BAKIYE) (6 . ,(- KOMA)))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Véase también

Glosario musical: Sección “sostenido” in *Glosario Musical*, Sección “bemol” in *Glosario Musical*, Sección “doble sostenido” in *Glosario Musical*, Sección “doble bemol” in *Glosario Musical*, Sección “Nombres de las notas” in *Glosario Musical*, Sección “semitono” in *Glosario Musical*.

Manual de aprendizaje: Sección “Accidentals and key signatures” in *Manual de Aprendizaje*.

Referencia de la notación: [Alteraciones accidentales automáticas], página 19, [Alteraciones de anotación (musica ficta)], página 292, [Nombres de las notas en otros idiomas], página 6.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Accidental_engraver” in *Referencia de Funcionamiento Interno*, Sección “Accidental” in *Referencia de Funcionamiento Interno*, Sección “AccidentalCautionary” in *Referencia de Funcionamiento Interno*, Sección “accidental-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No existen estándares reconocidos ampliamente para denotar los bemoles de tres cuartos, de manera que los símbolos de LilyPond no se ajustan a ningún estándar.

Nombres de las notas en otros idiomas

Existen conjuntos predefinidos de nombres de notas y sus alteraciones para algunos idiomas aparte del inglés. Para usarlos, incluya el archivo de inicio específico del idioma según aparece relacionado más abajo. Por ejemplo, para usar los nombres de las notas en español, escriba `\include "espanol.ly"` al principio del archivo de entrada.

Los archivos de idioma disponibles y los nombres de las notas que definen son los siguientes:

Archivo de idioma	**Nombre de las notas**
‘`nederlands.ly`’	c d e f g a bes b
‘`arabic.ly`’	do re mi fa sol la sib si
‘`catalan.ly`’	do re mi fa sol la sib si
‘`deutsch.ly`’	c d e f g a b h
‘`english.ly`’	c d e f g a bf b
‘`espanol.ly`’	do re mi fa sol la sib si
‘`italiano.ly`’	do re mi fa sol la sib si
‘`norsk.ly`’	c d e f g a b h
‘`portugues.ly`’	do re mi fa sol la sib si
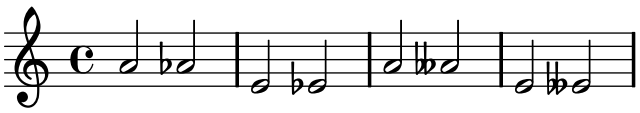
‘suomi.ly’	c d e f g a b h
‘svenska.ly’	c d e f g a b h
‘vlaams.ly’	do re mi fa sol la sib si

y los sufijos de las alteraciones correspondientes que definen son:

Archivo de idioma	sostenido	bemol	doble sostenido	doble bemol
‘nederlands.ly’	-is	-es	-isis	-eses
‘arabic.ly’	-d	-b	-dd	-bb
‘catalan.ly’	-d/-s	-b	-dd/-ss	-bb
‘deutsch.ly’	-is	-es	-isis	-eses
‘english.ly’	-s/-sharp	-f/-flat	-ss/-x/-sharpsharp	-ff/-flatflat
‘espanol.ly’	-s	-b	-ss	-bb
‘italiano.ly’	-d	-b	-dd	-bb
‘norsk.ly’	-iss/-is	-ess/-es	-ississ/-isis	-essess/-eses
‘portugues.ly’	-s	-b	-ss	-bb
‘suomi.ly’	-is	-es	-isis	-eses
‘svenska.ly’	-iss	-ess	-ississ	-essess
‘vlaams.ly’	-k	-b	-kk	-bb

En holandés, `aes` se contrae como `as`, pero las dos formas se aceptan en LilyPond. De forma similar, se aceptan tanto `es` como `ees`. Esto se aplica también a `aeses` / `ases` y a `eeses` / `eses`. A veces se definen solamente estos nombres contraídos en los archivos de idioma correspondientes.

```
a2 as e es a ases e eses
```



Algunas músicas utilizan microtonos cuyas alteraciones son fracciones de un sostenido o bemol ‘normales’. Los nombres de nota para cuartos de tono que se definen en los distintos archivos de idioma se encuentran listados en la tabla siguiente. Aquí los prefijos ‘semi-’ y ‘sesqui-’ significan ‘medio’ y ‘uno y medio’ respectivamente. Para los idiomas noruego, sueco, catalán y español no se ha definido aún ningún nombre especial.

Archivo de idioma	semi-sostenido	semi-bemol	sesqui-sostenido	sesqui-bemol
‘nederlands.ly’	-ih	-eh	-isih	-eseh
‘arabic.ly’	-sd	-sb	-dsd	-bsb
‘deutsch.ly’	-ih	-eh	-isih	-eseh
‘english.ly’	-qs	-qf	-tqs	-tqf
‘italiano.ly’	-sd	-sb	-dsd	-bsb
‘portugues.ly’	-sqt	-bqt	-stqt	-btqt

Véase también

Glosario musical: Sección “Nombres de las notas” in *Glosario Musical*.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

1.1.2 Modificación de varias notas a la vez
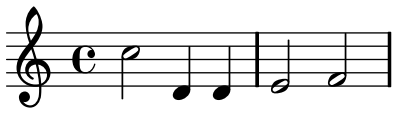
Esta sección trata de la manera de modificar las alturas.

Comprobación de octava

En el modo relativo, es fácil olvidar una marca de cambio de octava. Las comprobaciones de octava hacen más fácil encontrar estos errores, mediante la presentación de una advertencia y corrigiendo la octava si la nota se encuentra en una octava distinta de lo esperado.
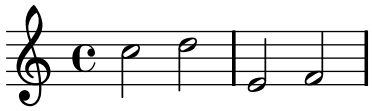
Para comprobar la octava de una nota, especifique la octava absoluta después del símbolo `=`. Este ejemplo genera un mensaje de advertencia (y corrige la altura) porque la segunda nota es la octava absoluta `d''` en lugar de `d'` como indica la corrección de octava.

```
\relative c'' {
  c2 d='4 d
  e2 f
}
```



La octava de las notas se puede comprobar también con la instrucción `\octaveCheck` *altura_de_control*. *altura_de_control* se especifica en modo absoluto. Esto comprueba que el intervalo entre la nota anterior y la *altura_de_control* se encuntra dentro de una cuarta (es decir, el cálculo normal para el modo relativo). Si esta comprobación fracasa, se imprime un mensaje de advertencia, pero la nota previa no se modifica. Las notas posteriores están en relación a la *altura_de_control*.
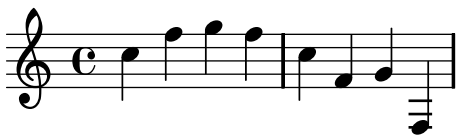
```
\relative c'' {
  c2 d
  \octaveCheck c'
  e2 f
}
```



Compare los dos compases siguientes. La primera y tercera comprobaciones de `\octaveCheck` fracasan, pero la segunda es correcto.

```
\relative c'' {
  c4 f g f

  c4
  \octaveCheck c'
  f
  \octaveCheck c'
  g
  \octaveCheck c'
  f
}
```



Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RelativeOctaveCheck” in *Referencia de Funcionamiento Interno*.

Transposición
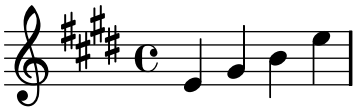
Una expresión musical se puede transportar mediante `\transpose`. La sintaxis es

```
\transpose nota_origen nota_destino expresión_musical
```

Esto significa que la *expresión_musical* se transporta el intervalo que hay entre las notas *nota_origen* y *nota_destino*: cualquier nota con la altura de `nota_origen` se cambia por `nota_destino` y cualquier otra nota se transporta el mismo intervalo. Las dos notas se introducen en modo absoluto.
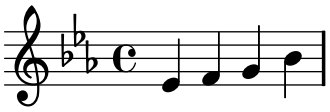
Por ejemplo, tomemos una pieza escrita en la tonalidad de Re mayor. Se puede transportar hacia arriba a Mi mayor; observe que también la armadura de la tonalidad se transporta automáticamente.

```
\transpose d e {
  \relative c' {
    \key d \major
    d4 fis a d
  }
}
```



Si una particella escrita en Do (*afinación de concierto* normal) se debe tocar con un clarinete en La (para el que un La se escribe como un Do, y que suena una tercera menor por debajo de lo que está escrito), la particella correspondiente se produce mediante:

```
\transpose a c' {
  \relative c' {
    \key c \major
    c4 d e g
  }
}
```
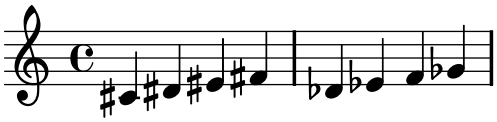


Observe que especificamos `\key c \major` de forma explícita. Si no especificamos ninguna tonalidad, las notas se transportan pero no se imprime la armadura.

`\transpose` distingue entre notas enarmónicas: tanto `\transpose c cis` como `\transpose c des` transportarán un semitono hacia arriba. La primera versión imprimirá sostenidos y las notas no se moverán de su lugar en la escala, la segunda imprimirá bemoles de la nota siguiente.

```
musica = \relative c' { c d e f }
\new Staff {
  \transpose c cis { \musica }
  \transpose c des { \musica }
```

```
}
```



`\transpose` también se puede usar para introducir notas escritas para un instrumento transpositor. Los ejemplos anteriores muestran cómo escribir alturas en Do (o en *afinación de concierto*) y tipografiarlas para un instrumento transpositor, pero también es posible el caso contrario si, por ejemplo, tenemos un conjunto de partes instrumentales y quiere hacer un guión en Do para el director. Por ejemplo, al introducir música para trompeta en Si bemol que comienza por un Mi en la partitura (Re de concierto), se puede escribir:

```
musicaEnSiBemol = { e4 ... }
\transpose c bes, \musicaEnSiBemol
```

Para imprimir esta música en Fa (por ejemplo, al arreglarla para trompa) puede envolver la música existente con otro `\transpose`:

```
musicaEnSiBemol = { e4 ... }
\transpose f c' { \transpose c bes, \musicaEnSiBemol }
```

Para ver más información sobre instrumentos transpositores, consulte [Transposición de los instrumentos], página 17.

Fragmentos de código seleccionados

Transportar música con el menor número de alteraciones

Este ejemplo utiliza código de Scheme para forzar las modificaciones enarmónicas de las notas, y así tener el menor número de alteraciones accidentales. En este caso se aplican las siguientes reglas:

- Se quitan las dobles alteraciones
- Si sostenido -> Do
- Mi sistenido -> Fa
- Do bemol -> Si
- Fa bemol -> Mi

De esta forma se selecciona el mayor número de notas enarmónicas naturales.

```
#(define  (naturalize-pitch p)
  (let* ((o (ly:pitch-octave p))
         (a (* 4 (ly:pitch-alteration p)))
    ; alteration, a, in quarter tone steps, for historical reasons
         (n (ly:pitch-notename p)))
    (cond
     ((and (> a 1) (or (eq? n 6) (eq? n 2)))
      (set! a (- a 2))
      (set! n (+ n 1)))
     ((and (< a -1) (or (eq? n 0) (eq? n 3)))
      (set! a (+ a 2))
      (set! n (- n 1))))
    (cond
     ((> a 2) (set! a (- a 4)) (set! n (+ n 1)))
     ((< a -2) (set! a (+ a 4)) (set! n (- n 1))))
    (if (< n 0) (begin (set! o (- o 1)) (set! n (+ n 7))))
    (if (> n 6) (begin (set! o (+ o 1)) (set! n (- n 7))))
```
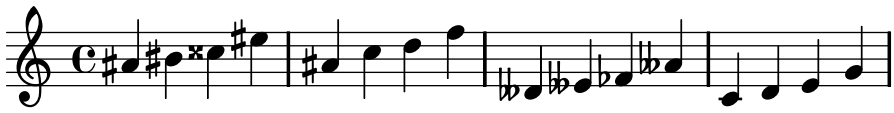
```
    (ly:make-pitch o n (/ a 4))))

#(define (naturalize musica)
  (let* ((es (ly:music-property musica 'elements))
         (e (ly:music-property musica 'element))
         (p (ly:music-property musica 'pitch)))
    (if (pair? es)
        (ly:music-set-property!
         musica 'elements
         (map (lambda (x) (naturalize x)) es)))
    (if (ly:music? e)
        (ly:music-set-property!
         musica 'element
         (naturalize e)))
    (if (ly:pitch? p)
        (begin
          (set! p (naturalize-pitch p))
          (ly:music-set-property! musica 'pitch p)))
    musica))

naturalizarMusica =
#(define-music-function (parser location m)
                                          (ly:music?)
                        (naturalize m))

musica = \relative c' { c4 d e g }

\score {
  \new Staff {
    \transpose c ais { \musica }
    \naturalizarMusica \transpose c ais { \musica }
    \transpose c deses { \musica }
    \naturalizarMusica \transpose c deses { \musica }
  }
  \layout { }
}
```



Véase también

Referencia de la notación: [Transposición de los instrumentos], página 17.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TransposedMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La conversión relativa no afecta a las secciones `\transpose`, `\chordmode` ni `\relative` dentro de su argumento. Para usar el modo relativo dentro de música transportada, se debe colocar otro `\relative` dentro de `\transpose`.
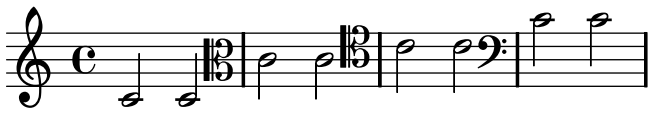
1.1.3 Imprimir las alturas

Esta sección trata de cómo alterar la presentación de la altura de las notas.

Clave

La clave se establece con la instrucción `\clef` *nombre_de_clave*. En todos los ejemplos se muestra el Do central.

```
\clef treble
c2 c
\clef alto
c2 c
\clef tenor
c2 c
\clef bass
c2 c
```
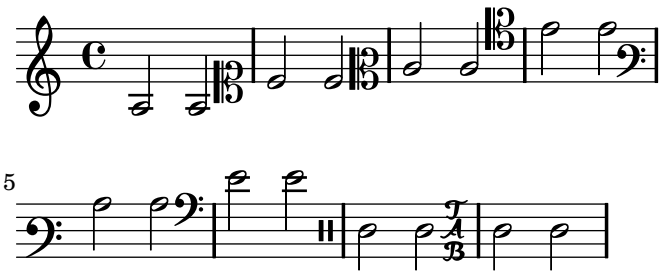


Otras claves son las siguientes:

```
\clef french
c2 c
\clef soprano
c2 c
\clef mezzosoprano
c2 c
\clef baritone
c2 c

\break
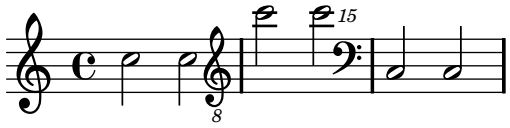
\clef varbaritone
c2 c
\clef subbass
c2 c
\clef percussion
c2 c
\clef tab
c2 c
```



Están contempladas claves adicionales que se describen bajo [Claves de la música mensural], página 287 y [Claves de canto gregoriano], página 295.

Al añadir `_8` o `^8` al nombre de la clave, la clave se transpone una octava hacia abajo o hacia arriba, respectivamente, y `_15` y `^15` la transpone dos octavas. El argumento *nombre_de_clave* se debe encerrar entre comillas cuando contiene guiones bajos o dígitos.

```
\clef treble
c2 c
\clef "treble_8"
c2 c
\clef "bass^15"
c2 c
```



Fragmentos de código seleccionados

Trucaje de las propiedades de clave

La instrucción `\clef "treble_8"` equivale a un ajuste de `clefGlyph`, `clefPosition` (que controla la posición vertical de la clave), `middleCPosition` y `clefOctavation`. Se imprime una clave cada vez que se modifica cualquiera de las propiedades excepto `middleCPosition`.

Observe que la modificación del glifo, la posición de la clave o su octavación, no cambian 'per se' la posición de las siguientes notas del pentagrama: para hacer esto también se debe especificar la posición del Do central. Los parámetros posicionales están en relación con la tercera línea del pentagrama, los números positivos desplazan hacia arriba, contando una unidad por cada línea y espacio. El valor de `clefOctavation` se establecería normalmente a 7, -7, 15 or -15, pero son válidos otros valores.

Cuando se produce un cambio de clave en el salto de línea se imprime la clave nueva tanto al final de la línea anterior como al principio de la nueva, de forma predeterminada. Si no se necesita la clave de advertencia al final de la línea anterior, se puede quitar estableciendo el valor de la propiedad `explicitClefVisibility` de `Staff`, a `end-of-line-invisible`. El comportamiento predeterminado se puede recuperar con `\unset Staff.explicitClefVisibility`.

Los siguientes ejemplos muestran las posibilidades cuando se ajustan estas propiedades manualmente. En la primera línea, los cambios manuales preservan el posicionamiento relativo estándar de las claves y las notas, pero no lo hacen en la segunda línea.

```
\layout { ragged-right = ##t }

{
  % Clave de sol predeterminada
  c'1
  % Clave de Fa estándar
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  c'1
  % Clave de barítono (do en quinta)
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  c'1
  % Clave estándar de tenor coral
```

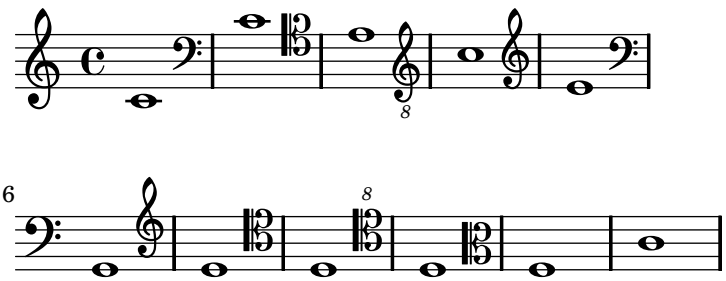
```
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefOctavation = #-7
  \set Staff.middleCPosition = #1
  c'1
  % Clave no estándar
  \set Staff.clefPosition = #0
  \set Staff.clefOctavation = #0
  \set Staff.middleCPosition = #-4
  c'1 \break

  % Los siguientes cambios de clave no preservan
  % la relación normal entre notas y claves:

  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'1
  \set Staff.clefGlyph = #"clefs.G"
  c'1
  \set Staff.clefGlyph = #"clefs.C"
  c'1
  \set Staff.clefOctavation = #7
  c'1
  \set Staff.clefOctavation = #0
  \set Staff.clefPosition = #0
  c'1

  % Aquí volvemos a la clave normal:

  \set Staff.middleCPosition = #0
  c'1
}
```



Véase también

Referencia de la notación: [Claves de la música mensural], página 287, [Claves de canto gregoriano], página 295.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Clef_engraver” in *Referencia de Funcionamiento Interno*, Sección “Clef” in *Referencia de Funcionamiento Interno*, Sección “OctavateEight” in *Referencia de Funcionamiento Interno*, Sección “clef-interface” in *Referencia de Funcionamiento Interno*.

Armadura de la tonalidad

Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios, porque las notas sin alteración pueden llevar signos de becuadro en función de la armadura. Para ver más información, consulte [Alteraciones accidentales], página 4 o Sección “Accidentals and key signatures” in *Manual de Aprendizaje*.
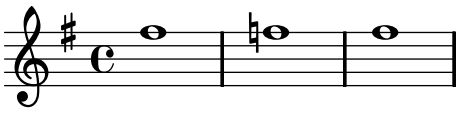
> **Nota:** Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios. En LilyPond, los nombres de las notas son el código de entrada en bruto; la armadura y la clave determinan de qué forma se presenta este código en bruto. Una nota sin alteración como `c` significa ‘Do natural’, si que le afecten la armadura ni la clave. Para ver más información, consulte Sección “Accidentals and key signatures” in *Manual de Aprendizaje*.

La armadura indica la tonalidad en que se toca una pieza. Está denotada por un conjunto de alteraciones (bemoles o sostenidos) al comienzo del pentagrama. El establecimiento o modificación de la armadura se hace con la instrucción `\key`:

```
\key nota tipo
```

Aquí, *tipo* debe ser `\major` o `\minor` para obtener la tonalidad *nota* mayor o *nota* menor, respectivamente. También puede usar los nombres estándar de modo (también conocidos como *modos eclesiásticos*): `\ionian` (jónico), `\dorian` (dórico), `\phrygian` (frigio), `\lydian` (lidio), `\mixolydian` (mixolidio), `\aeolian` (eolio) y `\locrian` (locrio).

```
\key g \major
fis1
f
fis
```
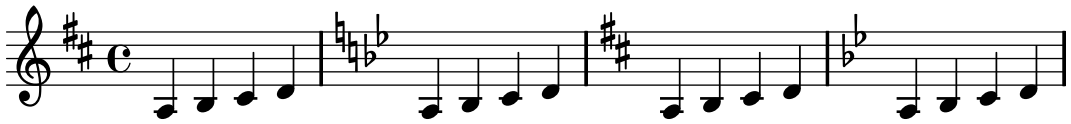


Fragmentos de código seleccionados

Evitar que se impriman becuadros cuando cambia la armadura

Cuando cambia la armadura de la tonalidad, se imprimen becuadros automáticamente para cancelar las alteraciones de las armaduras anteriores. Esto se puede evitar estableciendo al valor "falso" la propiedad `printKeyCancellation` del contexto `Staff`.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Armaduras de tonalidad no tradicionales

La muy utilizada instrucción `\key` establece la propiedad `keySignature` property, dentro del contexto `Staff`.
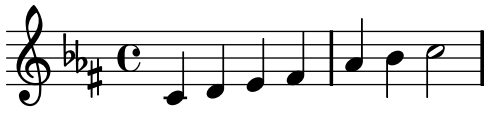
Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

`\set Staff.keySignature = #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)` donde, para cada elemento dentro de la lista, `octava` especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), `paso` especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y `alteración` es `,SHARP ,FLAT ,DOUBLE-SHARP` etc. (observe la coma precedente.)

Alternativamente, para cada elemento de la lista el uso del formato más conciso `(paso . alteración)` especifica que la misma alteración debe estar en todas las octavas.

He aquí un ejemplo de una posible armadura para generar una escala exátona:

```
\relative c' {
  \set Staff.keySignature = #`(((0 . 3) . ,SHARP)
                              ((0 . 5) . ,FLAT)
                              ((0 . 6) . ,FLAT))
  c4 d e fis
  aes4 bes c2
}
```



Véase también

Glosario musical: Sección “modo eclesiástico” in *Glosario Musical*, Sección “scordatura” in *Glosario Musical*.

Manual de aprendizaje: Sección “Accidentals and key signatures” in *Manual de Aprendizaje*.
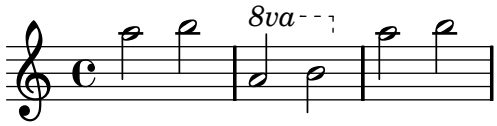
Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “KeyChangeEvent” in *Referencia de Funcionamiento Interno*, Sección “Key_engraver” in *Referencia de Funcionamiento Interno*, Sección “Key_performer” in *Referencia de Funcionamiento Interno*, Sección “KeyCancellation” in *Referencia de Funcionamiento Interno*, Sección “KeySignature” in *Referencia de Funcionamiento Interno*, Sección “key-cancellation-interface” in *Referencia de Funcionamiento Interno*, Sección “key-signature-interface” in *Referencia de Funcionamiento Interno*.

Corchetes de octava

Los *corchetes de Ottava* introducen un trasporte adicional de una octava para el pentagrama.

```
a'2 b
\ottava #1
a b
\ottava #0
a b
```



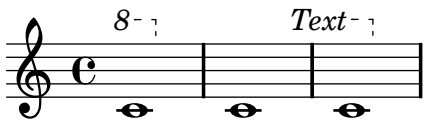
La función `ottava` también admite los valores -1 (para 8va bassa), 2 (para 15ma) y -2 (para 15ma bassa) como argumentos.

Fragmentos de código seleccionados

Texto de octava alta y baja

Internamente, la función `set-octavation` establece las propiedades `ottavation` (por ejemplo, a `"8va"` o a `"8vb"`) y `middleCPosition`. Para sobreescribir el texto del corchete, ajuste `ottavation` despues de invocar a `set-octavation`.

```
{
  \ottava #1
  \set Staff.ottavation = #"8"
  c''1
  \ottava #0
  c'1
  \ottava #1
  \set Staff.ottavation = #"Text"
  c''1
}
```



Véase también

Glosario musical: Sección “octavación” in *Glosario Musical*.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Ottava_spanner_engraver” in *Referencia de Funcionamiento Interno*, Sección “OttavaBracket” in *Referencia de Funcionamiento Interno*, Sección “ottava-bracket-interface” in *Referencia de Funcionamiento Interno*.

Transposición de los instrumentos

Al tipografiar partituras donde participan instrumentos transpositores, ciertas partes se pueden tipografiar en un tono distinto del *tono de concierto*. En estos casos, se debe especificar la tonalidad del *instrumento transpositor*; de otro modo, la salida MIDI y las notas guía en otras partes producirían alturas incorrectas. Para ver más información sobre partes citadas como guía, consulte [Citar otras voces], página 148.

`\transposition` *pitch*

El tono usado para `\transposition` debe corresponderse con el sonido real que se oye cuando el instrumento transpositor interpreta un Do central `c'` escrito en el pentagrama. Esta nota se escribe en altura absoluta, por tanto un instrumento que produce un sonido real un tono más agudo que la música impresa (un instrumento en Re) debe usar `\transposition d'`. La instrucción `\transposition` se debe usar *solamente* si las notas *no* se van a escribir en afinación de concierto.
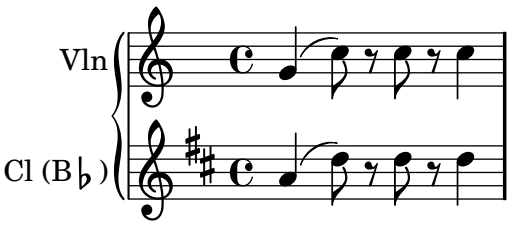
A continuación pueden verse algunas notas para violín y para clarinete en Si bemol, donde las partes se han introducido usando las notas y la armadura tal y como aparecen en la partitura del director. Lo que tocan los dos instrumentos está sonando al unísono.

```
\new GrandStaff <<
  \new Staff = "violin" {
    \relative c'' {
```

```
      \set Staff.instrumentName = #"Vln"
      \set Staff.midiInstrument = #"violin"
      % no es imprescindible pero es bueno recordarlo
      \transposition c'

      \key c \major
      g4( c8) r c r c4
    }
  }
  \new Staff = "clarinete" {
    \relative c'' {
      \set Staff.instrumentName = \markup { Cl (B\flat) }
      \set Staff.midiInstrument = #"clarinete"
      \transposition bes

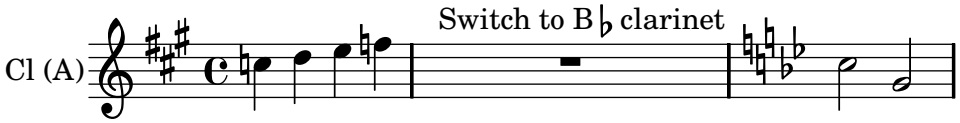
      \key d \major
      a4( d8) r d r d4
    }
  }
>>
```



La `\transposition` se puede cambiar durante la pieza. Por ejemplo, un clarinetista puede cambiar del clarinete en La al clarinete en Si bemol.

```
\set Staff.instrumentName = #"Cl (A)"
\key a \major
\transposition a
c d e f
\textLengthOn
s1*0^\markup { Switch to B\flat clarinet }
R1

\key bes \major
\transposition bes
c2 g
```



Véase también

Glosario musical: Sección “altura de concierto” in *Glosario Musical*, Sección “instrumento transpositor” in *Glosario Musical*.

Referencia de la notación: [Citar otras voces], página 148, [Transposición], página 9.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Alteraciones accidentales automáticas

Existen muchas convenciones distintas sobre la forma de tipografiar las alteraciones. LilyPond proporciona una función para especificar qué estilo de alteraciones usar. Esta función se invoca como sigue:

```
\new Staff <<
  #(set-accidental-style 'voice)
  { ... }
>>
```

El estilo de alteraciones se aplica al `Staff` en curso de forma predeterminada (con la excepción de los estilos `piano` y `piano-cautionary`, que se explican más adelante). Opcionalmente, la función puede tomar un segundo argumento que determina en qué ámbito se debe cambiar el estilo. Por ejemplo, para usar el mismo estilo en todos los pentagramas del `StaffGroup` en curso, use

```
#(set-accidental-style 'voice 'StaffGroup)
```

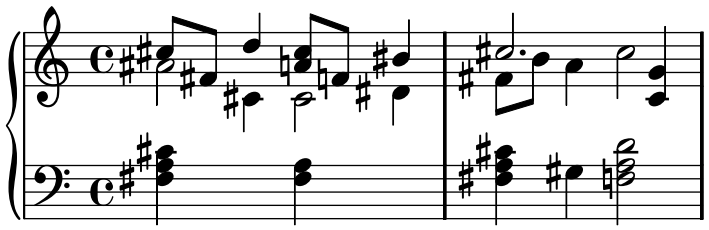
Están contemplados los siguientes estilos de alteración. Para dar una muestra de cada uno de los estilos, utilizamos el ejemplo siguiente:

```
musicaA = {
  <<
    \relative c' {
      cis'8 fis, d'4 <a cis>8 f bis4 |
      cis2. <c, g'>4 |
    }
    \\
    \relative c' {
      ais'2 cis, |
      fis8 b a4 cis2 |
    }
  >>
}

musicaB = {
  \clef bass
  \new Voice {
    \voiceTwo \relative c' {
      <fis, a cis>4
      \change Staff = arriba
      cis'
      \change Staff = abajo
      <fis, a>
      \change Staff = arriba
      dis' |
      \change Staff = abajo
      <fis, a cis>4 gis <f a d>2 |
    }
  }
}

\new PianoStaff {
  <<
    \context Staff = "arriba" {
```

```
        #(set-accidental-style 'default)
        \musicaA
      }
      \context Staff = "abajo" {
        #(set-accidental-style 'default)
        \musicaB
      }
    >>
  }
```

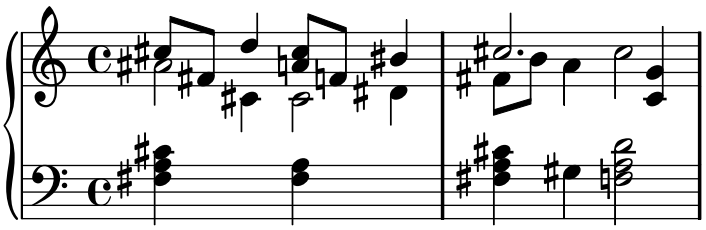


Observe que las últimas líneas de este ejemplo se pueden sustituir por las siguientes, siempre y cuando queramos usar el mismo estilo en los dos pentagramas.

```
  \new PianoStaff {
    <<
      \context Staff = "up" {
        %%% cambie la línea siguiente como desee:
        #(set-accidental-style 'default 'Score)
        \musicA
      }
      \context Staff = "down" {
        \musicB
      }
    >>
  }
```

`default (predeterminado)`
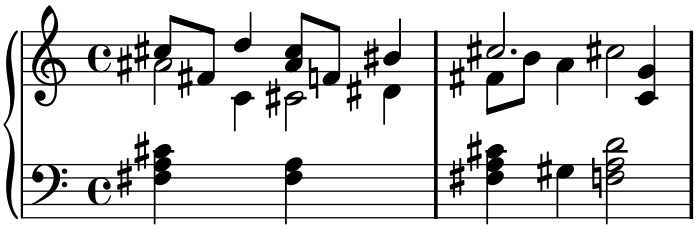
Es el comportamiento de composición tipográfica predeterminado. Corresponde a la práctica común del s. XVIII: las alteraciones accidentales se recuerdan hasta el final del compás en el que aparecen y sólo en la misma octava. Así, en el ejemplo siguiente, no se imprimen becuadros antes del Si natural en el segundo compás ni en el último Do:



`voice (voz)`

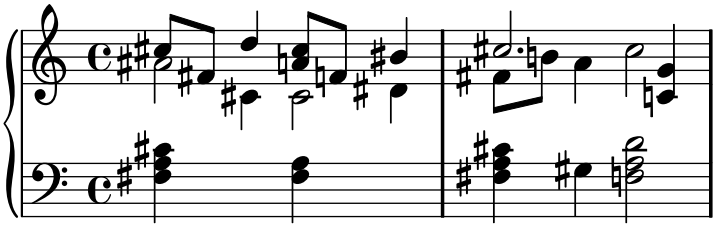
El comportamiento normal es recordar las alteraciones accidentales al nivel de `Staff`. En este estilo, sin embargo, se tipografían las alteraciones individualmente para cada voz. Aparte de esto, la regla es similar a `default`.

Como resultado, las alteraciones de una voz no se cancelan en las otras voces, lo que con frecuencia lleva a un resultado no deseado: en el ejemplo siguientem es difífil determinar si el segundo La se debe tocar natural o sostenido. Por tanto, la opción `voice` se debe usar sólo si las voces se van a leer individualmente por músicos distintos. Si el pentagrama va a utilizarse por parte de un solo músico (p.ej., un director, o en una partitura de piano), entonces se deben usar en su lugar los estilos `modern` o `modern-cautionary`.



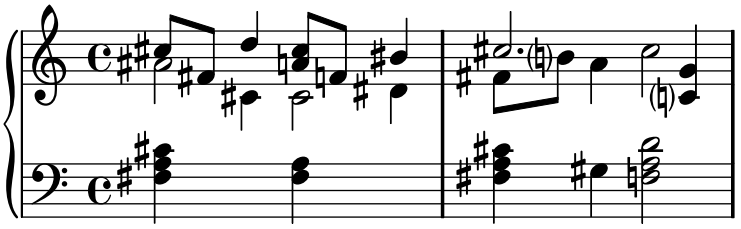
`modern (moderno)`

Esta regla corresponde a la práctica común del s. XX. Imprime las mismas alteraciones que el estilo `default`, con dos excepciones que sirven para evitar la ambigüedad: después de alteraciones temporales se imprimen indicaciones de cancelación también en el compás siguiente (para notas en la misma octava) y, en el mismo compás, para notas en octavas distintas. De aquí los becuadros antes del Si natural y del Do en el segundo compás del pentagrama superior:



`modern-cautionary (moderno de precaución)`

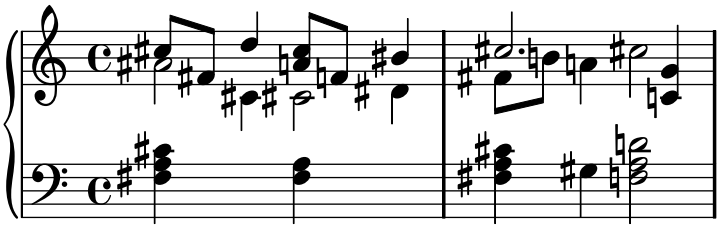
Esta regla es similar a `modern`, pero las alteraciones 'añadidas' (las que no se imprimen en el estilo `default`) se imprimen como alteraciones de precacución. Se imprimen de forma predeterminada con paréntesis, pero también se pueden imprimir en tamaño reducido definiendo la propiedad `cautionary-style` de `AccidentalSuggestion`.



`modern-voice (moderno, para voces)`
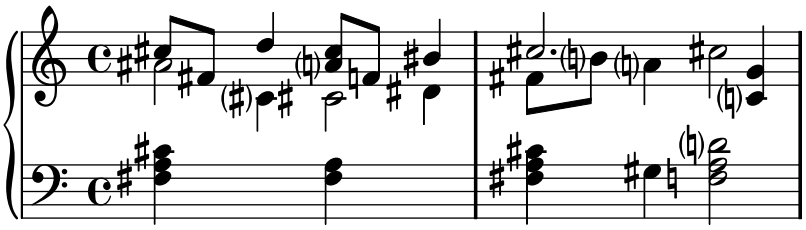
Esta regla se usa para que puedan leer las alteraciones en varias voces, tanto músicos que tocan una voz como músicos que tocan todas las voces. Se imprimen las alteraciones para cada voz, pero *se cancelan* entre voces dentro del mismo `Staff`. Por tanto, el La en el último compás se cancela porque la cancelación anterior estaba en

una voz distinta, y el Re en el pentagrama inferior se cancela a causa de la alteración en otra voz en el compás previo:



`modern-voice-cautionary (moderno, voz, de precaución)`

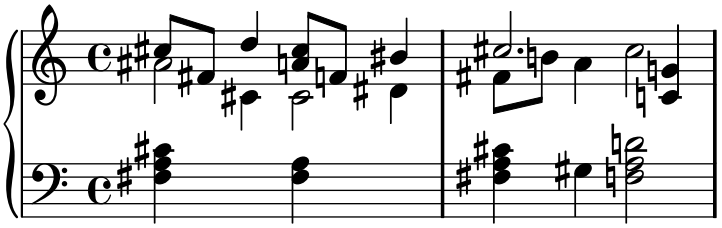
Esta regla es la misma que `modern-voice`, pero con las alteraciones añadidas (las que el estilo `voice` no imprime) compuestas como de precaución. Incluso aunque todas las alteraciones impresas por el estilo `default` *son* impresas con esta regla, algunas de ellas se tipografían como de precaución.



`piano`

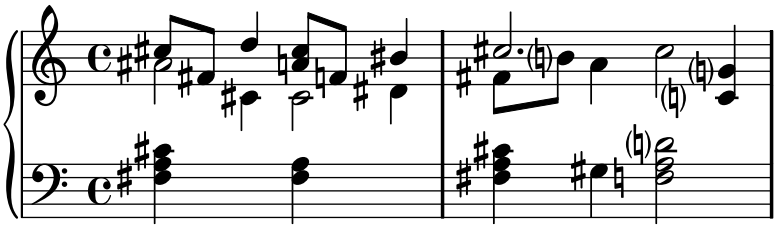
Esta regla refleja la práctica del s.XX para la notación de piano. Su comportamiento es muy similar al estilo `modern`, pero aquí las alteraciones también se cancelan entre distintos pentagramas del mismo grupo `GrandStaff` o `PianoStaff`, de ahí todas las cancelaciones de las últimas notas.

Este estilo de alteración se aplica de manera predeterminada al grupo `GrandStaff` o `PianoStaff` en curso.
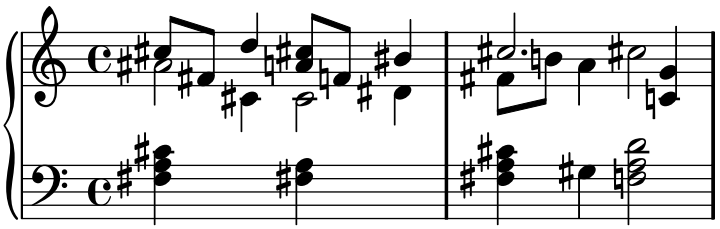


`piano-cautionary (piano, de precaución)`

Igual que `#(set-accidental-style 'piano)` pero con las alteraciones añadidas compuestas como de precaución.



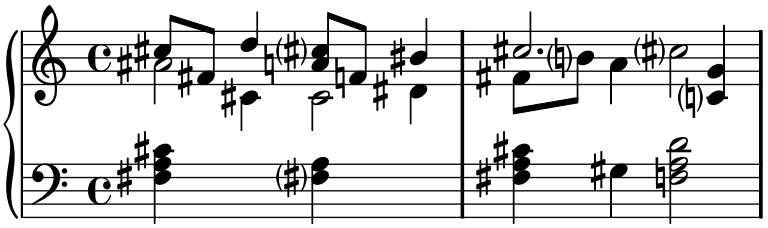
`neo-modern`

Esta regla reproduce una práctica común en la música contemporánea: las alteraciones accidentales se imprimen como en `modern`, pero se vuelven a imprimir si aparece la misma nota otra vez en el mismo compás (excepto si la nota se repite inmediatamente).
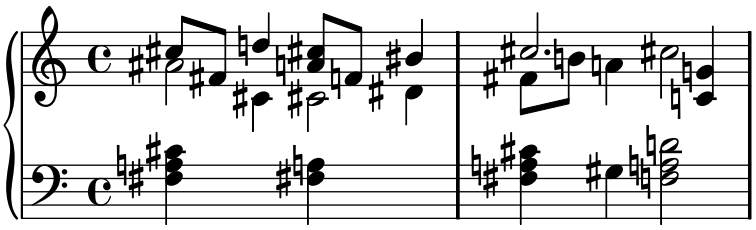


`neo-modern-cautionary`

Esta regla es similar a `neo-modern`, pero las alteraciones adicionales se imprimen como alteraciones de precaución.
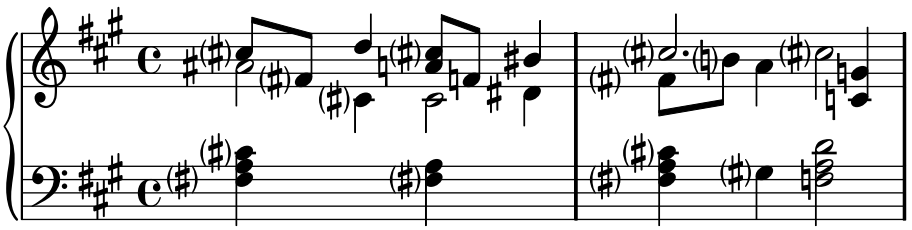


`dodecaphonic`

Esta regla refleja una práctica introducida por los compositores de principios del s.XX, en un intento de abolir la jerarquía entre notas naturales y alteradas. Con este estilo, *todas* las notas llevan alteración, incluso becuadros.
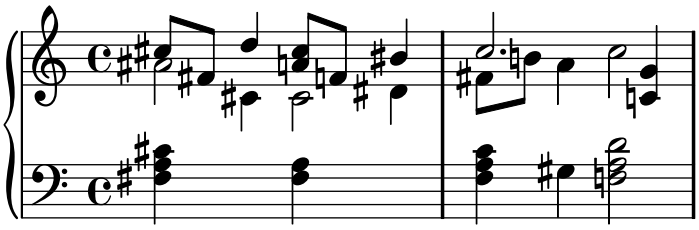


`teaching (enseñanza)`
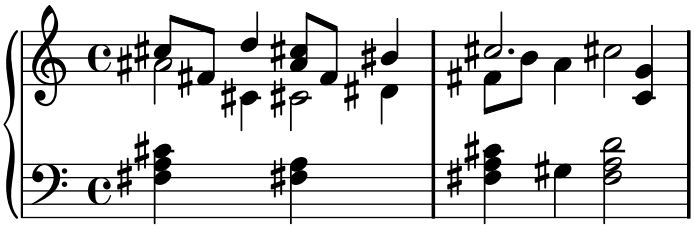
Esta regla está pensada para estudiantes, y hace más sencillo crear hojas de escalas con alteraciones de precaución creadas automáticamente. Las alteraciones se imprimen como en el estilo `modern`, pero se añaden alteraciones de precaución para todas las notas sostenidas o bemoles especificadas por la armadura, excepto si la nota se repite inmediatamente.



`no-reset (no restablecer)`
: Es el mismo que `default` pero con alteraciones que duran 'para siempre' y no sólo dentro del mismo compás:



`forget (olvidar)`
: Es lo opuesto a `no-reset`: Las alteraciones no se recuerdan en absoluto: de aquí que todas las alteraciones se compongan tipográficamente en relación a la armadura de la tonalidad, sin que estén afectadas por lo que viene antes. A diferencia de `dodecaphonic`, esta regla nunca imprime becuadros:
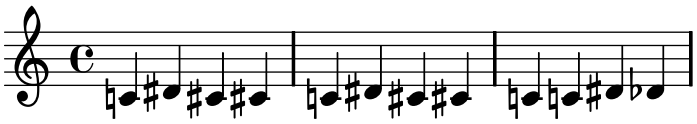


Fragmentos de código seleccionados

Alteraciones de estilo dodecafónico para todas las notas, incluidas las naturales

En las obras de principios del s.XX, empezando por Schoenberg, Berg y Webern (la "Segunda" escuela de Viena), cada nota de la escala de doce tonos se debe tratar con igualdad, sin ninguna jerarquía como los grados clásicos tonales. Por tanto, estos compositores imprimen una alteración accidental para cada nota, incluso en las notas naturales, para enfatizar su nuevo enfoque de la teoría y el lenguaje musicales.

Este fragmento de código muestra cómo conseguir dichas reglas de notación.

```
\score {
  \new Staff {
    #(set-accidental-style 'dodecaphonic)
    c'4 dis' cis' cis'
    c'4 dis' cis' cis'
    c'4 c' dis' des'
  }
  \layout {
    \context {
    \Staff
    \remove "Key_engraver"
    }
  }
}
```



Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Accidental” in *Referencia de Funcionamiento Interno*, Sección “Accidental_engraver” in *Referencia de Funcionamiento Interno*, Sección “GrandStaff” in *Referencia de Funcionamiento Interno* and Sección “PianoStaff” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “AccidentalSuggestion” in *Referencia de Funcionamiento Interno*, Sección “AccidentalPlacement” in *Referencia de Funcionamiento Interno*, Sección “accidental-suggestion-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Las notas simultáneas se consideran introducidas en modo secuencial. Esto significa que dentro de un acorde las alteraciones se tipografían como si las notas del acorde se sucedieran una a una en el tiempo, en el orden en que aparecen en el archivo de entrada. Esto plantea un problema cuando las alteraciones de un acorde dependen unas de otras, lo que no se da en el estilo ds alteraciones predeterminado. El problema se puede resolver insertando manualmente `!` y `?` para las notas problemáticas.
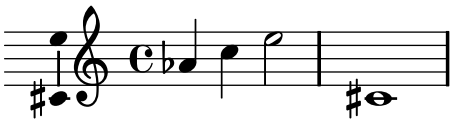
Tesitura

El término *ambitus* o ámbito, denota el rango de notas que abarca una voz dada en una parte musical. También puede denotar el margen de notas que es capaz de tocar un determinado instrumento musical. Los ámbitos se imprimen en las partes vocales de tal manera que los intérpretes puedan determinar con facilidad si cumplen con sus propias posibilidades.

Los ámbitos se presentan al comienzo de la pieza junto a la clave inicial. El rango se especifica gráficamente mediante dos cabezas de nota que representan a las notas inferior y superior. Sólo se imprimen alteraciones si no forman parte de la armadura de la tonalidad.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\relative c'' {
  aes c e2
  cis,1
}
```
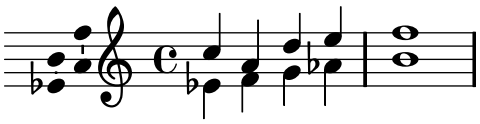


Fragmentos de código seleccionados

Añadir un ámbito por voz

Se puede añadir un ámbito por cada voz. En este caso, el ámbito se debe desplazar manualmente para evitar colisiones.
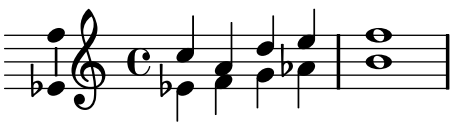
```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus #'X-offset = #2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Ámbitos con varias voces

La adición del grabador `Ambitus_engraver` al contexto de `Staff` crea un solo ámbito por pentagrama, incluso en el caso de pentagramas con varias voces.

```
\new Staff \with {
  \consists "Ambitus_engraver"
  }
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Véase también

Glosario musical: Sección “tesitura” in *Glosario Musical*.

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Ambitus_engraver” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “Ambitus” in *Referencia de Funcionamiento Interno*, Sección “AmbitusAccidental” in *Referencia de Funcionamiento Interno*, Sección “AmbitusLine” in *Referencia de Funcionamiento Interno*, Sección “AmbitusNoteHead” in *Referencia de Funcionamiento Interno*, Sección “ambitus-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No se efectúa un tratamiento de las posibles colisiones en caso de varias indicaciones de ámbito simultáneas en distintas voces.
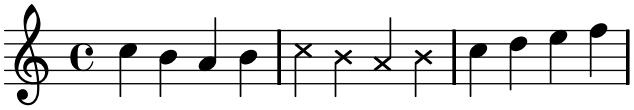
1.1.4 Cabeza de las notas

Esta sección propone formas de alterar las cabezas de las figuras.

Cabezas de nota especiales
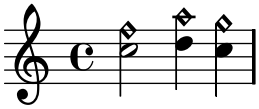
Se puede modificar la cabeza de las notas:

```
c4 b a b
\override NoteHead #'style = #'cross
c4 b a b
\revert NoteHead #'style
c4 d e f
```



Existe una abreviatura para las formas en rombo que sólo se puede usar en acordes:

```
<c f\harmonic>2 <d a'\harmonic>4 <c g'\harmonic>
```



Para ver todos los estilos de cabeza de las notas, consulte Sección B.7 [Estilos de cabezas de nota], página 475.

Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

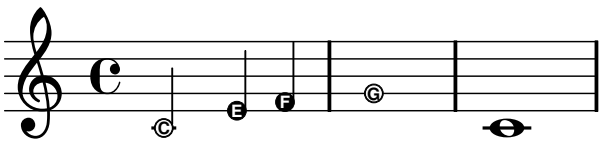
Referencia de la notación: Sección B.7 [Estilos de cabezas de nota], página 475.

Referencia de funcionamiento interno: Sección “note-event” in *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “Ledger_line_engraver” in *Referencia de Funcionamiento Interno*, Sección “NoteHead” in *Referencia de Funcionamiento Interno*, Sección “LedgerLineSpanner” in *Referencia de Funcionamiento Interno*, Sección “note-head-interface” in *Referencia de Funcionamiento Interno*, Sección “ledger-line-spanner-interface” in *Referencia de Funcionamiento Interno*.

Cabezas de notas de Notación Fácil

Las cabezas 'easy play' (fáciles de tocar) tienen el nombre de la nota (en inglés) dentro de la cabeza. Se usa en la música para principiantes.

```
#(set-global-staff-size 26)
\relative c' {
  \easyHeadsOn
  c2 e4 f
  g1
  \easyHeadsOff
  c,1
}
```



La instrucción `\easyHeadsOn` sobreescribe los ajustes del objeto `NoteHead`. Estos ajustes se pueden deshacer mediante la isntrucción `\easyHeadsOff`. Para que las letras sean legibles, se tienen que imprimir en un tamaño grande de la tipografía. Para imprimir con un tamaño de tipografía mayor, consulte Sección 4.2.1 [Establecer el tamaño del pentagrama], página 351.

Instrucciones predefinidas

`\easyHeadsOn`, `\easyHeadsOff`.

Véase también

Referencia de la notación: Sección 4.2.1 [Establecer el tamaño del pentagrama], página 351.
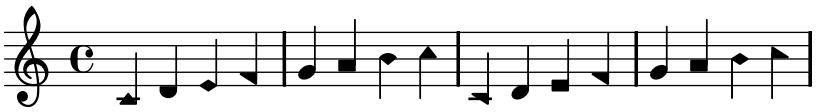
Fragmentos de código: Sección "Pitches" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "note-event" in *Referencia de Funcionamiento Interno*, Sección "Note_heads_engraver" in *Referencia de Funcionamiento Interno*, Sección "NoteHead" in *Referencia de Funcionamiento Interno*, Sección "note-head-interface" in *Referencia de Funcionamiento Interno*.

Cabezas de notas con formas diversas

En la notación de cabezas con forma, la forma de la cabeza corresponde a la función armónica de una nota dentro de la escala. Esta notación se hizo popular en los libros americanos de canciones durante el s.XIX.

```
\aikenHeads
c, d e f g a b c
\sacredHarpHeads
c, d e f g a b c
```



Las formas se determinan en función del grado de la escala, donde la tónica está determinada por la instrucción `\key`

Instrucciones predefinidas

`\aikenHeads`, `\sacredHarpHeads`.

Fragmentos de código seleccionados

Aplicar estilos de cabeza según la nota de la escala

La propiedad `shapeNoteStyles` se puede usar para definir varios estilos de cabezas de nota para cada grado de la escala (según esté establecido por la armadura o por la propiedad "tonic"). Esta propiedad requiere un conjunto de símbolos, que pueden ser puramente arbitrarios (se permiten expresiones geométricas como `triangle`, triángulo, `cross`, aspas, y `xcircle`, círculo con aspas) o basados en una antigua tradición americana de grabado (ciertos nombres de nota latinos trambién se permiten).

Dicho esto, para imitar antiguos cancioneros americanos, existen varios estilos predefinidos de cabezas de nota disponibles a través de instrucciones de abreviatura como `\aikenHeads` o `\sacredHarpHeads`.

Este ejemplo muestra distintas formas de obtener cabezas de notas con forma, y muestra la capacidad de transportar una melodía sin perder la correspondencia entre las funciones armónicas y los estilos de cabezas de nota.
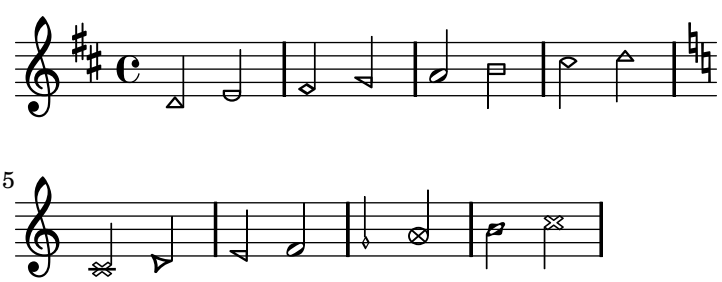
```
\layout { ragged-right = ##t }

fragmento = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\score {
  \new Staff {
    \transpose c d
    \relative c' {
      \set shapeNoteStyles = #'#(do re mi fa
                                 #f la ti)
      \fragmento
    }

    \break

    \relative c' {
      \set shapeNoteStyles  = #'#(cross triangle fa #f
                                  mensural xcircle diamond)
      \fragmento
    }
  }
}
```



Para ver todos los estilos de formas de cabeza de nota, consulte Sección B.7 [Estilos de cabezas de nota], página 475.

Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

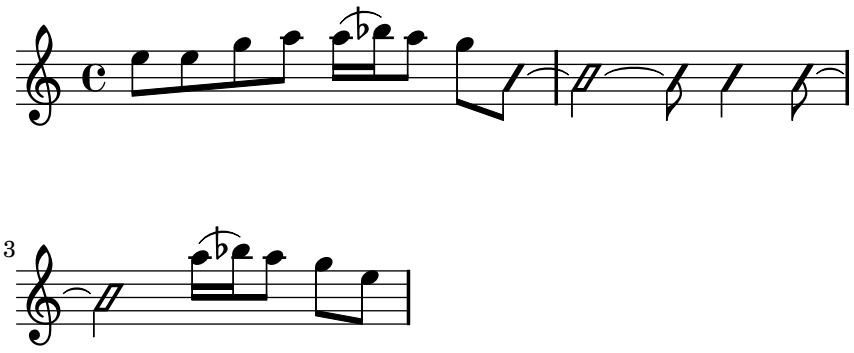
Referencia de la notación: Sección B.7 [Estilos de cabezas de nota], página 475.

Referencia de funcionamiento interno: Sección “note-event” in *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “NoteHead” in *Referencia de Funcionamiento Interno*, Sección “note-head-interface” in *Referencia de Funcionamiento Interno*.

Improvisación

La improvisación se denota a veces mediante cabezas de nota en forma de barra inclinada, donde el ejecutante puede elegir cualquier nota pero con el ritmo especificado. Estas cabezas de nota se crean así:

```
\new Voice \with {
  \consists "Pitch_squash_engraver"
} {
  e8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  e2 ~ e8 f4 f8 ~
  f2
  \improvisationOff
  a16( bes) a8 g e
}
```



Instrucciones predefinidas
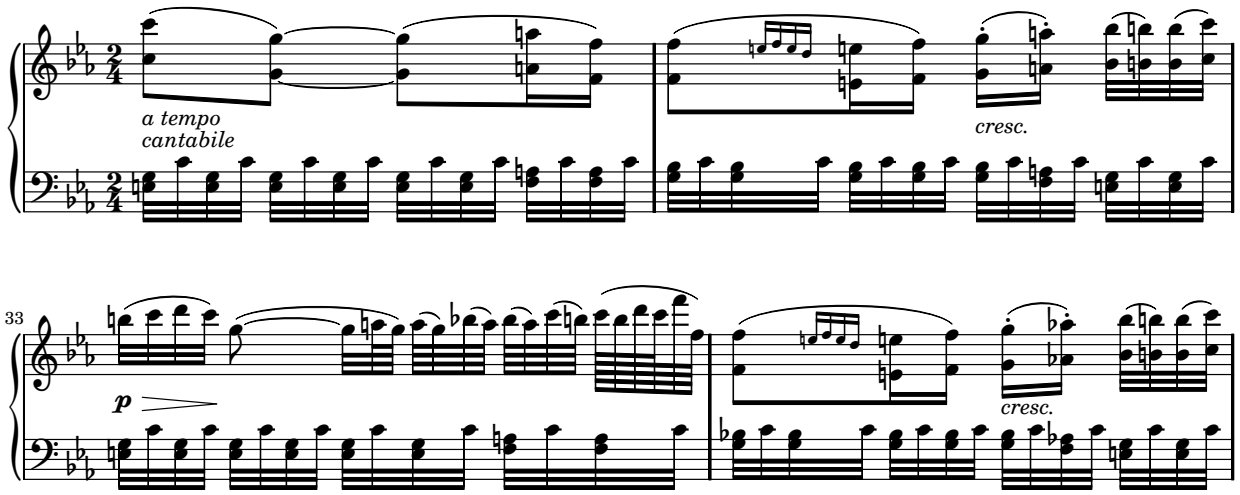
`\improvisationOn`, `\improvisationOff`.

Véase también

Fragmentos de código: Sección “Pitches” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Pitch_squash_engraver” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*, Sección “RhythmicStaff” in *Referencia de Funcionamiento Interno*.

1.2 Duraciones



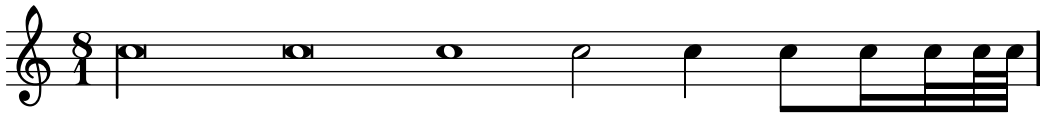
Esta sección trata de los ritmos, los silencios, las duraciones, las barras y los compases.

1.2.1 Escritura de las duraciones (valores rítmicos)

Duraciones
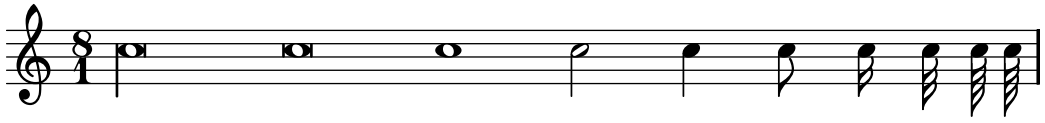
Las duraciones se designan mediante números y puntos. Las duraciones se introducen como sus valores recíprocos respecto de la redonda. Por ejemplo, una negra se escribe usando un `4` (puesto que es 1/4 de redonda), mientras que una blanca se escribe con un `2` (por ser 1/2 de redonda). Para notas mayores de la redonda se deben usar los comandos `\longa` (que es una breve doble) y `\breve`. Se pueden especificar duraciones tan cortas como la garrapatea (con el número 128). Son posibles valores más cortos, pero sólo como notas unidas por una barra.

```
\time 8/1
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c64
```



Aquí se pueden ver las mismas duraciones con el barrado automático desactivado.

```
\time 8/1
\autoBeamOff
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c64
```



Se puede escribir una nota con la duración de una cuádruple breve mediante `\maxima`, pero esto está contemplado solamente dentro de la notación musical antigua. Para ver más detalles, consulte Sección 2.8 [Notación antigua], página 283.
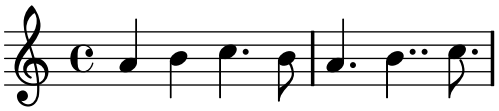
Si la duración se omite, su valor será el de la duración de la nota anterior. Por omisión, el valor de la primera nota es el de una negra.

```
a a a2 a a4 a a1 a
```



Para obtener duraciones de notas con puntillo, escriba un punto (`.`) después del número. Las notas con doble puntillo se especifican escribiendo dos puntos, y así sucesivamente.

```
a4 b c4. b8 a4. b4.. c8.
```



Ciertas duraciones no se pueden representar sólo con duraciones binarias y puntillos; la única forma de representarlas es ligando dos o más notas. Para ver más detalles, consulte [Ligaduras de unión], página 36.

Para ver de qué manera se pueden especificar las duraciones de las sílabas de la letra y cómo alinear la letra a las notas, consulte Sección 2.1 [Música vocal], página 190.

Opcionalmente se pueden espaciar las notas de forma estrictamente proporcional a sus duraciones. Para ver más detalles sobre este y otros ajustes que controlan la notación proporcional, consulte Sección 4.5.5 [Notación proporcional], página 378.

Normalmente los puntillos se desplazan hacia arriba para evitar las líneas del pentagrama, excepto en situaciones de polifonía. Para forzar una dirección concreta de forma manual, véase Sección 5.4.2 [Dirección y posición], página 407.

Instrucciones predefinidas

`\autoBeamOff`, `\dotsUp`, `\dotsDown`, `\dotsNeutral`.

Véase también

Glosario musical: Sección “breve” in *Glosario Musical*, Sección “longa” in *Glosario Musical*, Sección “figura” in *Glosario Musical*, Sección “Nombre de las duraciones de notas y silencios” in *Glosario Musical*.

Referencia de la notación: [Barras automáticas], página 56, [Ligaduras de unión], página 36, Sección 1.2.1 [Escritura de las duraciones (valores rítmicos)], página 31, Sección 1.2.2 [Escritura de silencios], página 39, Sección 2.1 [Música vocal], página 190, Sección 2.8 [Notación antigua], página 283, Sección 4.5.5 [Notación proporcional], página 378.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Dots” in *Referencia de Funcionamiento Interno*, Sección “DotColumn” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No existe un límite fundamental para las duraciones de los silencios (tanto para el más largo como para el más corto), pero el numero de glifos es limitado: sólo se pueden imprimir desde el silencio de garrapatea (128) hasta el de máxima (8 redondas).
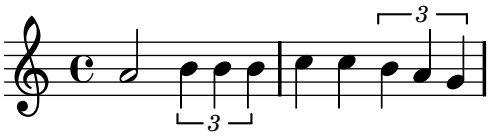
Grupos especiales

Los grupos especiales se obtienen a partir de una expresión musical multiplicando todas las duraciones por una fracción:

```
\times fracción { música }
```

La duración de *música* se multiplicará por la fracción. El denominador de la fracción se imprimirá sobre las notas, opcionalmente con un corchete. El grupo especial más común es el tresillo, en el que 3 notas tienen el valor de 2, por tanto las notas duran 2/3 de su valor escrito
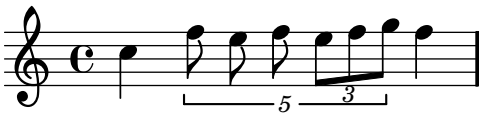
```
a2 \times 2/3 { b4 b b }
c4 c \times 2/3 { b4 a g }
```



La colocación automática del corchete de grupo especial por encima o por debajo de las notas se puede sobreescribir manualmente mediante instrucciones predefinidas, para ver más detalles consulte Sección 5.4.2 [Dirección y posición], página 407.

Los grupos pueden anidarse unos dentro de otros:

```
\autoBeamOff
c4 \times 4/5 { f8 e f \times 2/3 { e[ f g] } } f4 |
```



La modificación de los grupos especiales anidados que comienzan en el mismo instante musical se debe hacer con `\tweak`.

Para modificar la duración de las notas sin imprimir un corchete de grupo especial, véase [Escalar las duraciones], página 35.

Instrucciones predefinidas

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

Fragmentos de código seleccionados

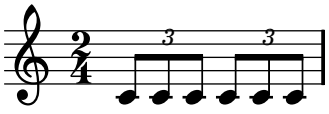
Escribir varios grupos especiales usando una sola instrucción \times

La propiedad `tupletSpannerDuration` establece cuánto debe durar cada grupo de valoración especial contenido dentro del corchete que aparece después de `\times`. Así, se pueden escribir muchos tresillos seguidos dentro de una sola expresión `\times`, ahorrando trabajo de teclado.

En el ejemplo se muestran dos tresillos, aunque se ha escrito `\times` una sola vez.

Para ver más inforamción sobre `make-moment`, véase "Administración del tiempo".

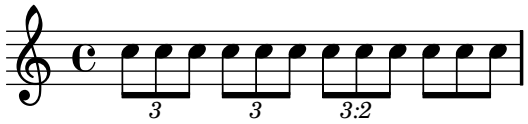
```
\relative c' {
  \time 2/4
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```



Cambiar el número del grupo especial

De forma predeterminada sólo se imprime el numerador del grupo especial sobre el corchete de grupo, es decir, el denominador del argumento de la instrucción `\times`. De forma alternativa, se puede imprimr un quebrado en la forma numerador:denominador del número del grupo, o eliminar el número.
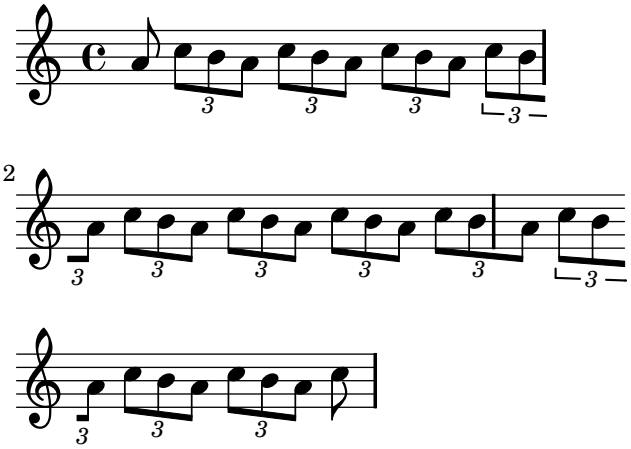
```
\relative c'' {
  \times 2/3 { c8 c c } \times 2/3 { c8 c c }
  \override TupletNumber #'text = #tuplet-number::calc-fraction-text
  \times 2/3 { c8 c c }
  \override TupletNumber #'stencil = ##f
  \times 2/3 { c8 c c }
}
```



Permitir saltos de línea dentro de grupos especiales con barra

Este ejemplo artificial muestra cómo se pueden permitir tanto los saltos de línea manuales como los automáticos dentro de un grupo de valoración especial unido por una barra. Observe que estos grupos sincopados se deben barrar manualmente.

```
\layout {
  \context {
    \Voice
    % Permitir saltos de línea dentro de grupos especiales
    \remove "Forbid_line_break_engraver"
    % Permitir romper barras en saltos de línea
    \override Beam #'breakable = ##t
  }
}
\relative c'' {
  a8
  \repeat unfold 8 { \times 2/3 { c[ b a] } }
  % Insertar salto de línea manual dentro de un grupo especial
  \times 2/3 { c[ b \bar "" \break a] }
  \repeat unfold 2 { \times 2/3 { c[ b a] } }
  c8
}
```



Véase también

Glosario musical: Sección “tresillo” in *Glosario Musical*, Sección “grupo especial” in *Glosario Musical*, Sección “polimétrica” in *Glosario Musical*.

Manual de aprendizaje: Sección “Tweaking methods” in *Manual de Aprendizaje*.

Referencia de la notación: [Gestión del tiempo], página 83, [Escalar las duraciones], página 35, Sección 5.3.4 [La instrucción tweak], página 403, [Notación polimétrica], página 50.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TupletBracket” in *Referencia de Funcionamiento Interno*, Sección “TupletNumber” in *Referencia de Funcionamiento Interno*, Sección “TimeScaledMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos
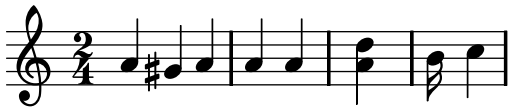
Cuando la primera nota de un pentagrama es una nota de adorno (un mordente de una nota) seguida de un grupo de valoración especial, la nota de adorno se debe situar antes de la instrucción `\times` para evitar errores. En cualquier otro lugar, las notas de adorno sí se pueden colocar dentro de los corchetes de grupo especial.

Escalar las duraciones

Es posible alterar la duración de las figuras, silencios o acordes en un factor `N/M` añadiendo `*N/M` (o `*N` si `M` es 1) a la duración. Esto no afectará a la apariencia de las notas o silencios que se producen, pero la duración alterada se usará para calcular la posición dentro del compás y para establecer la duración en la salida MIDI. Los factores de multiplicación se pueden combinar en la forma `*L*M/N`.

En el siguiente ejemplo las tres primeras notas duran exactamente dos partes, pero no se imprime ningún corchete de tresillo.

```
\time 2/4
% Alterar la duración de los tresillos
a4*2/3 gis4*2/3 a4*2/3
% Duraciones normales
a4 a4
% Duplicar la duración del acorde
<a d>4*2
% Duración de negra, aparece como semicorchea
b16*4 c4
```
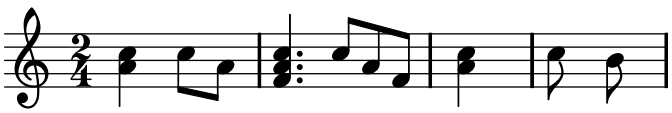


La duración de las notas de salto o espaciadoras también se puede modificar mediante un multiplicador. Esto es útil para saltar muchos compases, como por ejemplo `s1*23`.

De la misma forma, se pueden comprimir por una fracción trozos de música más largos, como si cada nota, acorde o silencio tuviera la fracción como multiplicador. Esto dejará intacta la apariencia de la *música*, pero la duración interna de las notas se multiplicará por la fracción *numerador*/*denominador*. Los espacios que rodean el punto son necesarios. He aquí un ejemplo que muestra cómo se puede comprimir y expandir la música:

```
\time 2/4
% Duraciones normales
<c a>4 c8 a
```

```
% Escalar la música en un factor 2/3
\scaleDurations #'(2 . 3) {
  <c a f>4. c8 a f
}
% Escalar la música en un factor de 2
\scaleDurations #'(2 . 1) {
  <c' a>4 c8 b
}
```



Una aplicación de esta instrucción se encuentra en la notación polimétrica, véase [Notación polimétrica], página 50.

Véase también

Referencia de la notación: [Grupos especiales], página 33, [Silencios invisibles], página 41, [Notación polimétrica], página 50.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Ligaduras de unión

Una ligadura de unión conecta dos notas adyacentes de la misma altura. La ligadura en efecto extiende la longitud de una nota.

> **Nota:** No deben confundirse las ligaduras de unión con las *ligaduras de expresión*, que indican articulación, ni con las *ligaduras de fraseo*, que indican el fraseo musical. Una ligadura de unión es tan sólo una manera de extender la duración de una nota, algo parecido a lo que hace el puntillo.
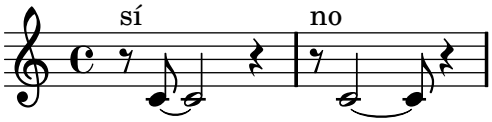
Se introduce una ligadura de unión usando el símbolo de la tilde curva ~

```
a2 ~ a
```



Se usan ligaduras de unión bien cuando la nota atraviesa la barra de compás o bien cuando no se pueden usar puntillos para denotar el ritmo. También se deben usar ligaduras cuando las notas atraviesan subdivisiones del compás de mayor duración:

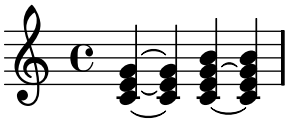
```
\relative c' {
  r8^"sí" c8 ~ c2 r4 |
  r8^"no" c2 ~ c8 r4
}
```



Si necesitamos ligar muchas notas a través de las líneas divisorias, nos podría resultar más fácil utilizar la división automática de las notas, véase [División automática de las notas], página 53. Este mecanismo divide automáticamente las notas largas y las liga a través de las barras de compás.
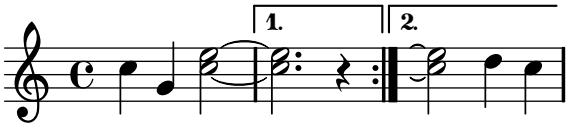
Cuando se aplica una ligadura de unión a un acorde, se conectan todas las cabezas de las notas cuyas alturas coinciden. Si no coincide ningún par de cabezas, no se crea ninguna ligadura. Los acordes se pueden ligar parcialmente colocando la ligadura dentro del acorde.
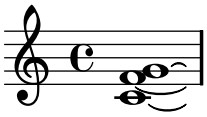
```
<c e g> ~ <c e g>
<c~ e g~ b> <c e g b>
```



Cuando la segunda vez de una repetición comienza con una nota ligada, es necesario especificar la ligadura repetida como sigue:

```
\repeat volta 2 { c g <c e>2 ~ }
\alternative {
  % Primera alternativa: la nota siguiente se liga normalmente
  { <c e>2. r4 }
  % Segunda alternativa, la nota siguiente lleva una ligadura repetida
  { <c e>2\repeatTie d4 c } }
```



Las ligaduras *L.v.* (*laissez vibrer*, dejar vibrar) indican que las notas no se deben apagar al final. Se usan en la notación para piano, arpa y otros instrumentos de cuerda y percusión. Se pueden introducir de la siguiente manera:

```
<c f g>1\laissezVibrer
```



Es posible controlar la colocación vertical de las ligaduras de unión; véase la sección de instrucciones predefinidas, o para ver más detalles, consulte Sección 5.4.2 [Dirección y posición], página 407.

Se pueden especificar ligaduras continuas, de puuntos o intermitentes, véase la sección de instrucciones predefinidas.

Instrucciones predefinidas
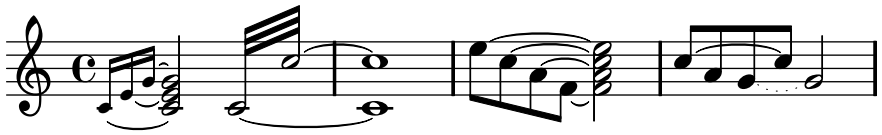
`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieSolid`, `\repeatTie`.

Fragmentos de código seleccionados

Uso de ligaduras en los arpegios

En ocasiones se usan ligaduras de unión para escribir los arpegios. En este caso, las dos notas ligadas no tienen que ser consecutivas. Esto se puede conseguir estableciendo la propiedad `tieWaitForNote` al valor "verdadero". La misma funcionalidad es de utilidad, por ejemplo, para ligar un trémolo a un acorde, pero en principio, también se puede usar para notas normales consecutivas, como se muestra en este ejemplo.
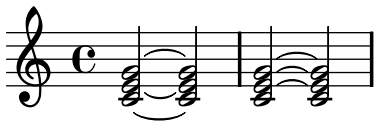
```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



Grabado manual de las ligaduras

Se pueden grabar a mano las ligaduras modificando la propiedad `tie-configuration` del objeto `TieColumn`. El primer número indica la distancia a partir de la tercera línea del pentagrama en espacios de pentagrama, y el segundo número indica la dirección (1 = hacia arriba, -1 = hacia abajo).

```
\relative c' {
  <c e g>2 ~ <c e g>
  \override TieColumn #'tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g> ~ <c e g>
}
```



Véase también

Glosario musical: Sección “ligadura de unión” in *Glosario Musical*, Sección “laissez vibrer” in *Glosario Musical*.

Referencia de la notación: [División automática de las notas], página 53.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “LaissezVibrerTie” in *Referencia de Funcionamiento Interno*, Sección “LaissezVibrerTieColumn” in *Referencia de Funcionamiento Interno*, Sección “TieColumn” in *Referencia de Funcionamiento Interno*, Sección “Tie” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Un cambio de pentagrama cuando hay una ligadura activa no producirá una ligadura inclinada.

Los cambios de clave o de octava durante una ligadura de unión no están bien definidos realmente. En estos casos puede ser preferible una ligadura de expresión.

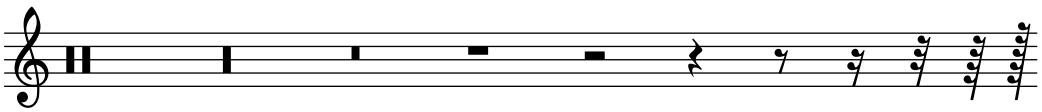
1.2.2 Escritura de silencios

Los silencios se escriben como parte de la música dentro de las expresiones musicales.

Silencios

Los silencios se introducen como notas con el nombre `r`. Las duraciones mayores que la redonda utilizan las instrucciones predefinidas que se muestran aquí:

```
\new Staff {
  % Estas dos líneas son solamente para embellecer el ejemplo
  \time 16/1
  \override Staff.TimeSignature #'stencil = ##f
  % Imprimir un silencio de máxima, equivalente a cuatro breves
  r\maxima
  % Imprimir un silencio de longa, igual a dos breves
  r\longa
  % Imprimir un silencio de breve
  r\breve
  r1 r2 r4 r8 r16 r32 r64 r128
}
```



Los silencios de un compás, centrados en medio del compás, se deben hacer con silencios multicompás. Se pueden usar para un solo compás así como para muchos compases, y se tratan en [Silencios de compás completo], página 42.

Para especificar explícitamente la posición vertical de un silencio, escriba una nota seguida de `\rest`. Se colocará un silencio en la posición en que debería aparecer la nota. Esto posibilita la aplicación manual precisa de formato a la música polifónica, ya que el formateador automático de colisiones de silencios no mueve estos silencios.

```
a4\rest d4\rest
```



Fragmentos de código seleccionados

Estilos de silencios

Los silencios se pueden imprimir en distintos estilos.

```
\layout {
  indent = 0.0
  \context {
    \Staff
    \remove "Time_signature_engraver"
```

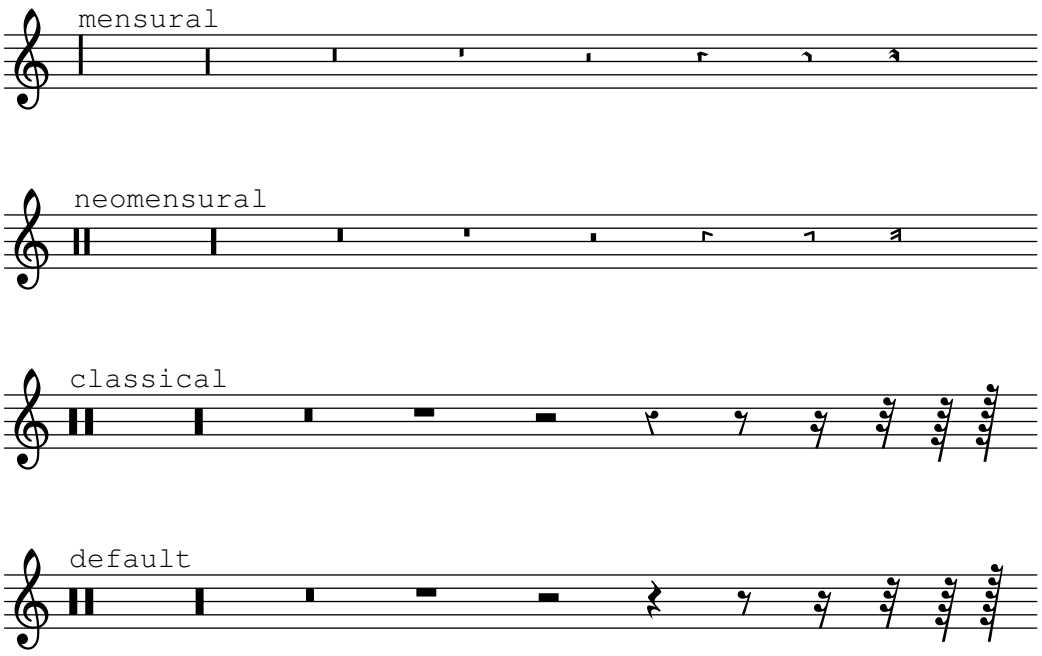
```
  }
}

\new Staff \relative c {
  \cadenzaOn
  \override Staff.Rest #'style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest #'style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest #'style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""

  \override Staff.Rest  #'style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}
```



Véase también

Referencia de la notación: [Silencios de compás completo], página 42.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Rest” in *Referencia de Funcionamiento Interno*.
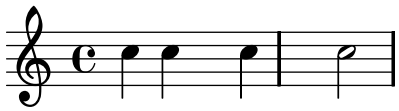
Advertencias y problemas conocidos

No existe un límite fundamental respecto de las duraciones de los silencios (tanto para el más corto como para el más largo), pero el número de glifos es limitado: hay silencios desde la garrapatea (128) hasta la máxima (8 redondas).

Silencios invisibles
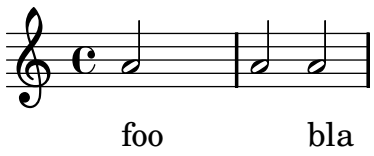
Un silencio invisible (también conocido como 'skip' o desplazamiento) se puede introducir como una nota con el nombre `s`:

```
c4 c s c
s2 c
```



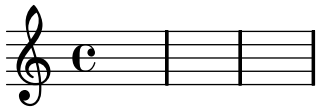
Los silencios de separación sólo están disponible en el modo de notas y en el modo de acordes. En otras situaciones, por ejemplo, cuando se introduce la letra, se usa la instrucción `\skip` para producir un desplazamiento de una cierta magnitud temporal. `\skip` requiere una duración explícita.

```
<<
  {
    a2 \skip2 a2 a2
  }
  \new Lyrics {
    \lyricmode {
      foo2 \skip 1 bla2
    }
  }
>>
```



Un silencio de separación produce implícitamente contextos `Staff` y `Voice` si no existe ninguno, igual que las notas y los silencios normales:

```
s1 s s
```



`\skip` tan sólo desplaza un tiempo musical; no produce ninguna salida, de ninguna clase.

```
% Esto es una entrada válida, pero no hace nada
\skip 1 \skip1 \skip 1
```

Véase también

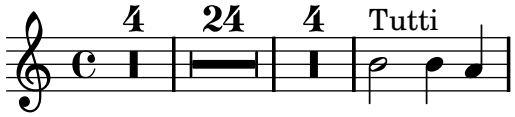
Fragmentos de código: Sección "Rhythms" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "SkipMusic" in *Referencia de Funcionamiento Interno*.

Silencios de compás completo

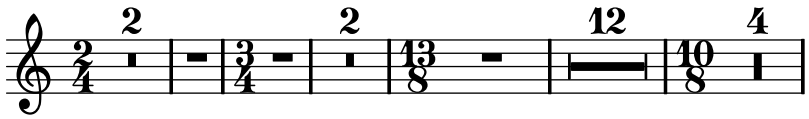
Los silencios de uno o más compases completos se introducen como notas con el nombre `R` en mayúscula:

```
% Compases en silencio contraídos a un solo compás
\compressFullBarRests
R1*4
R1*24
R1*4
b2^"Tutti" b4 a4
```
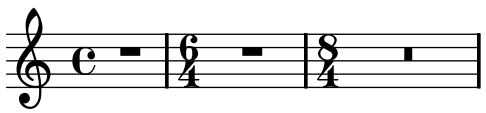


La duración de los silencios de compás completo es idéntica a la notación de la duración que se usa para las notas. La duración de un silencio multi-compás debe ser siempre un número entero de compases, por lo que con frecuencia han de utilizarse puntillos o fracciones:

```
\compressFullBarRests
\time 2/4
R1 | R2 |
\time 3/4
R2. | R2.*2 |
\time 13/8
R1*13/8 | R1*13/8*12 |
\time 10/8
R4*5*4 |
```



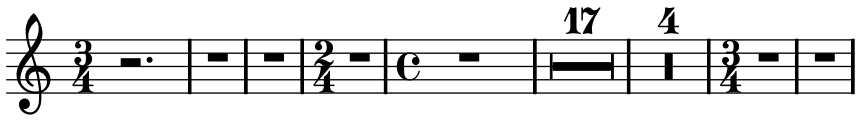
Un silencio de un compás completo se imprime como un silencio de redonda o de breve, centrado en el compás, según el tipo de compás vigente.

```
\time 4/4
R1 |
\time 6/4
R1*3/2 |
\time 8/4
R1*2 |
```



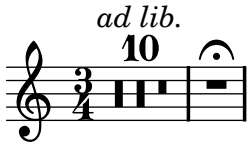
De forma predeterminada un silencio multicompás se expande en la partitura impresa para mostrar explícitamente todos los compases de silencio. De forma alternatica, un silencio multicompás se puede presentar como un solo compás que contiene un símbolo de silencio multicompás, con el número de compases impreso encima del compás:

```
% Comportamiento predeterminado
\time 3/4 r2. | R2.*2 |
\time 2/4 R2 |
\time 4/4
% Compases en silencio contraídos a un solo compás
\compressFullBarRests
r1 | R1*17 | R1*4 |
% Compases de silencio expandidos
\expandFullBarRests
\time 3/4
R2.*2 |
```



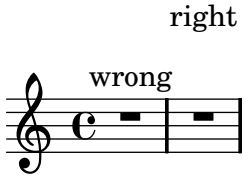
Se pueden añadir elementos de marcado a los silencios multicompás. Se proporciona la instrucción predefinida `\fermataMarkup` para añadir calderones.

```
\compressFullBarRests
\time 3/4
R2.*10^\markup { \italic "ad lib." }
R2.^\fermataMarkup
```



> **Nota:** Los elementos de marcado que se añaden a un silencio multicompás son objetos del tipo `MultiMeasureRestText`, no `TextScript`. Las sobreescrituras de propiedades deben ir dirigidas hacia el objeto correcto, o se ignorarán. Véase el ejemplo siguiente.

```
% Esto no funciona, se ha especificado un nombre de objeto equivocado
\override TextScript #'padding = #5
R1^"wrong"
% Esto es correcto y funciona
\override MultiMeasureRestText #'padding = #5
R1^"right"
```



Cuando un silencio multicompás sigue inmediatamente al establecimiento de un compás parcial con `\partial`, es posible que no se emitan las advertencias correspondientes de comprobación de compás.

Instrucciones predefinidas
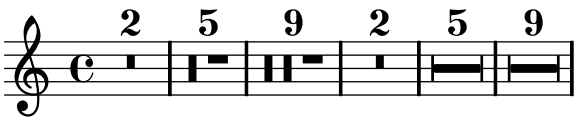
`\textLengthOn`, `\textLengthOff`, `\fermataMarkup`, `\compressFullBarRests`, `\expandFullBarRests`.

Fragmentos de código seleccionados

Cambiar la forma de los silencios multicompás

Si hay diez compases de silencio o menos, se imprime en el pentagrama una serie de silencios de breve y longa (conocidos en alemán como "Kirchenpausen", «silencios eclesiásticos»); en caso contrario se muestra una barra normal. Este número predeterminado de diez se puede cambiar sobreescribiendo la propiedad `expand-limit`:

```
\relative c'' {
  \compressFullBarRests
  R1*2 | R1*5 | R1*9
  \override MultiMeasureRest #'expand-limit = #3
  R1*2 | R1*5 | R1*9
}
```

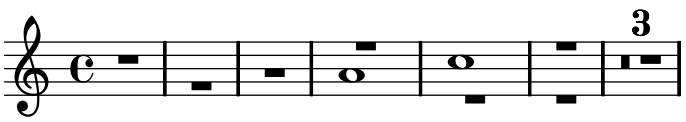


Posicionar los silencios multicompás

A diferencia de los silencios normales, no existe una instrucción predefinida para modificar la posición predefinida de un símbolo multicompás sobre el pentagrama, adjuntándolo a una nota, independientemente de cuál sea su forma. Sin embargo, en la música polifónica los silencios multicompás de las voces de numeración par e impar están separados verticalmente. La colocación de los silencios multicompás se puede controlar como se ve a continuación:

```
\relative c'' {
  % Los silencios multicompás se sitúan de forma predeterminada bajo la segunda línea
  R1
  % Se pueden mover con un override
  \override MultiMeasureRest #'staff-position = #-2
  R1
  % Un valor de cero es la posición predeterminada;
  % el truco siguiente mueve el silencio a la línea central
  \override MultiMeasureRest #'staff-position = #-0.01
  R1
  % Los silencios multicompás de las voces impares están bajo la línea superior
  << { R1 } \\ { a1 } >>
  % Los silencios multicompás de las voces pares están bajo la línea inferior
  << { c1 } \\ { R1 } >>
  % Siguen estando separados incluso en compases vacíos
  << { R1 } \\ { R1 } >>
  % Así los juntamos incluso si hay dos voces
  \compressFullBarRests
  <<
    \revert MultiMeasureRest #'staff-position
    { R1*3 }
    \\
    \revert MultiMeasureRest #'staff-position
```

```
    { R1*3 }
  >>
}
```

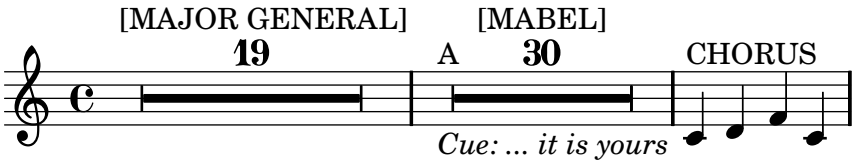


Marcado de silencios multicompás

Los elementos de marcado aplicados a un silencio multicompás se centran encima o debajo de éste. Los elementos de marcado extensos que se adjuntan a silencios multicompás no producen la expansión del compás. Para expandir un silencio multicompás de forma que quepa todo el marcado, utilice un silencio de separación con un marcado aplicado antes del silencio multicompás.

Observe que el silencio separador produce la inserción de un compás. El texto aplicado a un siencio sparador de esta forma se alinea por la izquierda a la posición en que la nota estaría situada dentro del compás, pero si la longitud del compás está determinada por la longitud del texto, éste aparecerá centrado.

```
\relative c' {
  \compressFullBarRests
  \textLengthOn
  s1*0^\markup { [MAJOR GENERAL] }
  R1*19
  s1*0_\markup { \italic { Cue: ... it is yours } }
  s1*0^\markup { A }
  R1*30^\markup { [MABEL] }
  \textLengthOff
  c4^\markup { CHORUS } d f c
}
```



Véase también

Glosario musical: Sección “silencios multicompás” in *Glosario Musical*.

Referencia de la notación: [Duraciones], página 31, Sección 1.8 [Texto], página 165, Sección 1.8.2 [Formatear el texto], página 173, [Guiones de texto], página 166.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “MultiMeasureRest” in *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestNumber” in *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestText” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Si se trata de usar digitaciones (p. ej., `R1*10-4`) para poner números encima de los silencios multicompás, la cifra de la digitación (el 4) puede chocar con el número del compás (aquí el 10).

No hay ninguna forma de condensar automáticamente muchos silencios en un solo silencio multicompás.
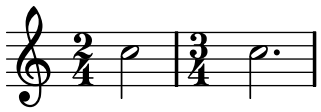
Los silencios multicompás no toman parte en las colisiones de silencios.

1.2.3 Impresión de las duraciones

Indicación de compás
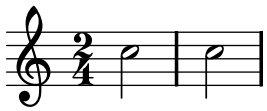
La indicación de compás se establece como sigue:
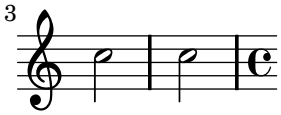
```
\time 2/4 c2
\time 3/4 c2.
```
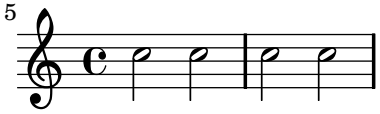


La indicación de compás se imprime al comienzo de una pieza y siempre que hay un cambio de compás. Si se produce un cambio al final de una línea, se imprime una indicación de advertencia en dicho lugar. Se puede modificar este comportamiento predeterminado, véase Sección 5.4.6 [Visibilidad de los objetos], página 415.

```
\time 2/4
c2 c
\break
c c
\break
\time 4/4
c c c c
```
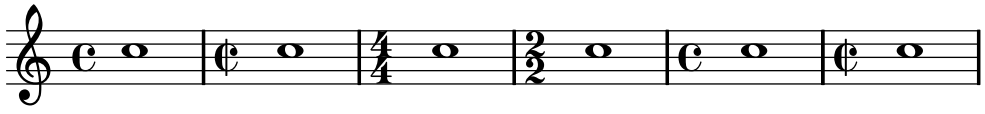






El símbolo de compás que se usa en 2/2 y 4/4 se puede cambiar a un estilo numérico:

```
% Estilo predeterminado
\time 4/4 c1
\time 2/2 c1
% Cambiar al estilo numérico
\numericTimeSignature
\time 4/4 c1
\time 2/2 c1
% Retornar al estilo predeterminado
\defaultTimeSignature
\time 4/4 c1
\time 2/2 c1
```



Las indicaciones de compás de la música mensural se tratan en [Indicaciones de compás de la música mensural], página 288.

Instrucciones predefinidas

`\numericTimeSignature`, `\defaultTimeSignature`.

Fragmentos de código seleccionados

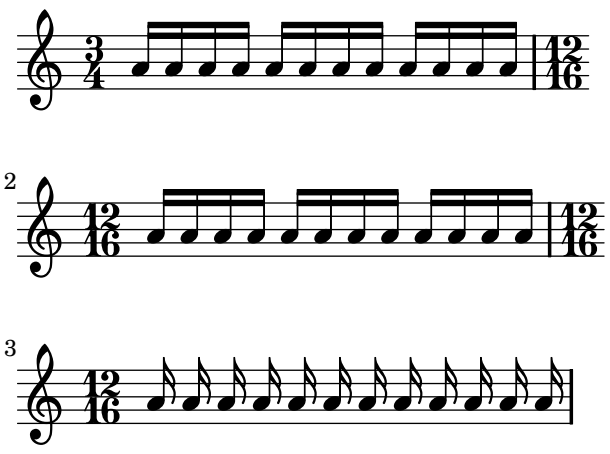
Cambio de compás sin afectar al barrado

La instrucción `ime` establece las propiedades `timeSignatureFraction`, `beatLength`, `beatGrouping` y `measureLength` en el contexto `Timing`, que normalmente tiene el alias `Score`. La modificación del valor de `timeSignatureFraction` hace que se imprima la nueva indicación de compás sin que cambie ninguna de las demás propiedades:

```
\relative c'' {
  \time 3/4
  a16 a a a a a a a a a a a

  % Cambio de compás manteniendo el barrado de 3/4
  % a causa de que el compás subyacente no ha cambiado
  \set Score.timeSignatureFraction = #'(12 . 16)
  a16 a a a a a a a a a a a

  \time 12/16
  % Perder el barrado de 3/4 ahora que se ha cambiado el \time
  a16 a a a a a a a a a a a
}
```
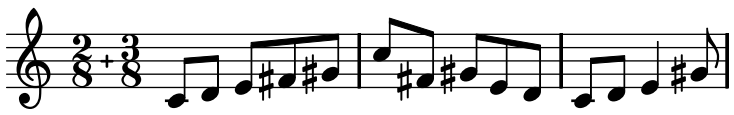


Indicaciones de compases compuestos

Las indicaciones de compás poco frecuentes como "5/8" se pueden ejecutar como compases compuestos (p.ej. "3/8 + 2/8"), que combinan dos o más metros diferentes. LilyPond puede hacer la música de este tipo fácil de leer e interpretar, imprimiendo explícitamente las indicaciones de compás compuesto y adaptando el comportamiento automático de las barras (también se pueden añadir indicaciones gráficas de la agrupación de compases; véase el fragmento de código apropiado en la base de datos).

```
#(define ((compound-time one two num) grob)
  (grob-interpret-markup grob
    (markup #:override '(baseline-skip . 0) #:number
      (#:line (
```

```
            (#:column (one num))
            #:vcenter "+"
            (#:column (two num))))
        )))

\relative c' {
  \override Staff.TimeSignature #'stencil = #(compound-time "2" "3" "8")
  \time 5/8
  #(override-auto-beam-setting '(end 1 8 5 8) 1 4)
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```



Véase también

Glosario musical: Sección “indicación de compás” in *Glosario Musical*

Referencia de la notación: [Indicaciones de compás de la música mensural], página 288, [Gestión del tiempo], página 83.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TimeSignature” in *Referencia de Funcionamiento Interno*, Sección “Timing_translator” in *Referencia de Funcionamiento Interno*.
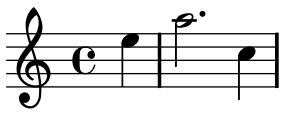
Anacrusas

Los compases parciales como las anacrusas o partes al alzar se escriben usando la instrucción `\partial`, con la sintaxis

```
\partial duración
```
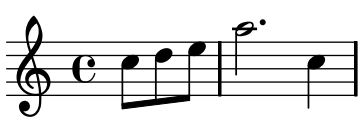
donde `duración` es el valor rítmico que se se debe añadir antes del comienzo del siguiente compás completo:

```
\partial 4 e4 |
a2. c,4 |
```



El compás parcial puede ser de cualquier duración menor de un compás completo:

```
\partial 8*3 c8 d e |
a2. c,4 |
```



Esto se traduce internamente a

```
\set Timing.measurePosition = -duración
```

La propiedad `measurePosition` contiene un número racional que indica qué parte del compás ha transcurrido hasta el momento. Observe que se establece a un número negativo por parte de la instrucción `\partial`: es decir, `\partial 4` se traduce internamente a `-4`, con el significado de "queda un valor de negra en este compás".

Véase también

Glosario musical: Sección "anacrusa" in *Glosario Musical*.

Referencia de la notación: [Notas de adorno], página 78.

Fragmentos de código: Sección "Rhythms" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Timing_translator" in *Referencia de Funcionamiento Interno*.
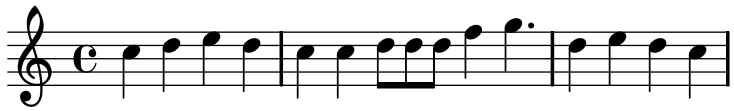
Advertencias y problemas conocidos

El uso de `\partial` está previsto solamente para el comienzo de una pieza. Si se emplea después del comienzo se obtendrán varias advertencias desagradables.

Música sin compasear
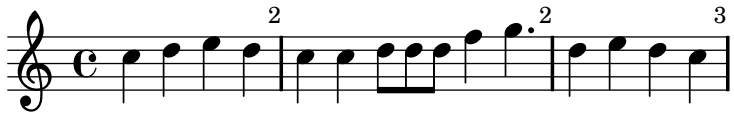
Las líneas divisorias y los números de compás se calculan automáticamente. Para música sin compasear (por ejemplo cadencias), esto no es deseable. Para desactivar las barras y los números de compás automáticos, utilice la instrucción `\cadenzaOn`, y para activarlos de nuevo use `\cadenzaOff`.

```
c4 d e d
\cadenzaOn
c4 c d8 d d f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```



La numeración de compases se continúa al final de la cadencia como si la cadencia no existiera:

```
% Mostrar todos los números de compás
\override Score.BarNumber #'break-visibility = #all-visible
c4 d e d
\cadenzaOn
c4 c d8 d d f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```



Instrucciones predefinidas

`\cadenzaOn`, `\cadenzaOff`.

Véase también

Glosario musical: Sección “cadenza” in *Glosario Musical*.

Referencia de la notación:

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Advertencias y problemas conocidos

LilyPond inserta saltos de línea y de página solamente en las barras de compás. A menos que la música sin compasear acabe antes del final de la línea del pentagrama, tendrá que insertar barras de compás invisibles

```
\bar ""
```

para indicar dónde se pueden producir los saltos.

Notación polimétrica

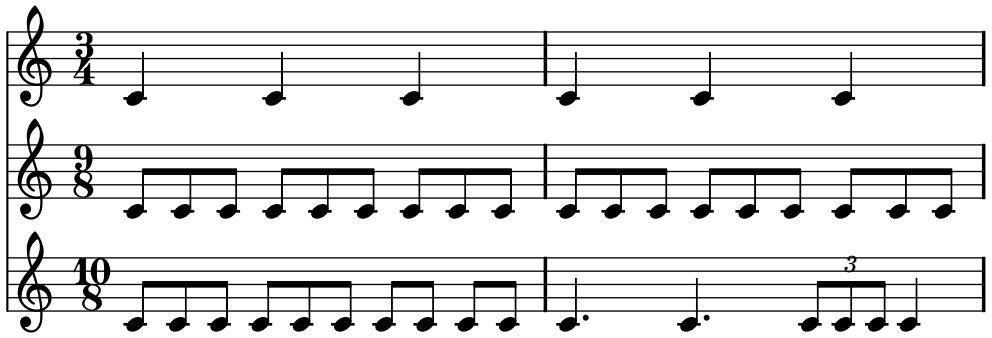
Está contemplada la notación polimétrica, ya sea explícitamente o mediante la modificación de la indicación visible del compás y el escalado de la duración de las notas.

Pentagramas con distintas indicaciones de compás, compases de igual longitud

Se puede crear esta notación estableciendo una indicación de compás común para cada pentagrama per sustituyendo el símbolo manualmente mediante el establecimiento de `timeSignatureFraction` a la fracción deseada y escalando las duraciones impresas en cada pentagrama a la indicación de compás común; véase [Indicación de compás], página 46. El escalado se hace con `\scaleDurations`, que se usa de una forma similar a `\times`, pero no crea un corchete de grupo especial, véase [Escalar las duraciones], página 35.

En este ejemplo, se usa en paralelo música con compases de 3/4, 9/8 y 10/8. En el segundo pentagrama, las duraciones mostradas se multiplican por 2/3, pues 2/3 * 9/8 = 3/4, y en el tercer pentagrama, las duraciones que se muestran están multiplicadas por 3/5, pues 3/5 * 10/8 = 3/4. Con frecuencia se hace necesario insertar las barras de forma manual, pues el escalado de las duraciones afecta a las reglas de barrado automático.

```
\relative c' <<
  \new Staff {
    \time 3/4
    c4 c c |
    c c c |
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = #'(9 . 8)
    \scaleDurations #'(2 . 3)
    \repeat unfold 6 { c8[ c c] }
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = #'(10 . 8)
    \scaleDurations #'(3 . 5) {
      \repeat unfold 2 { c8[ c c] }
      \repeat unfold 2 { c8[ c] } |
      c4. c4. \times 2/3 { c8[ c c] } c4
    }
  }
>>
```



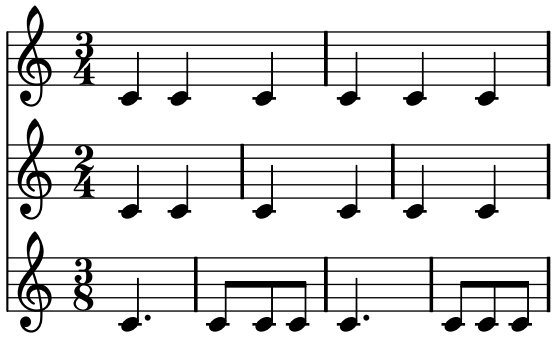
Pentagramas con distinas indicaciones de compás y longitudes de compás distintas

Se puede dar a cada pentagrama su propia indicación de compás independiente trasladando los grabadores `Timing_translator` y `Default_bar_line_engraver` al contexto de `Staff`.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

% Ahora cada pentagrama tiene un compás distinto

\relative c' <<
  \new Staff {
    \time 3/4
    c4 c c |
    c c c |
  }
  \new Staff {
    \time 2/4
    c4 c |
    c c |
    c c |
  }
  \new Staff {
    \time 3/8
    c4. |
    c8 c c |
    c4. |
    c8 c c |
  }
>>
```
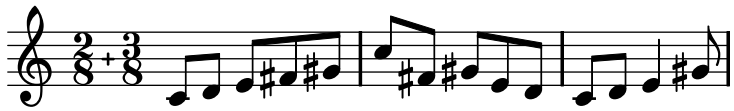


Fragmentos de código seleccionados

Indicaciones de compases compuestos

Las indicaciones de compás poco frecuentes como "5/8" se pueden ejecutar como compases compuestos (p.ej. "3/8 + 2/8"), que combinan dos o más metros diferentes. LilyPond puede hacer la música de este tipo fácil de leer e interpretar, imprimiendo explícitamente las indicaciones de compás compuesto y adaptando el comportamiento automático de las barras (también se pueden añadir indicaciones gráficas de la agrupación de compases; véase el fragmento de código apropiado en la base de datos).

```
#(define ((compound-time one two num) grob)
  (grob-interpret-markup grob
    (markup #:override '(baseline-skip . 0) #:number
      (#:line (
          (#:column (one num))
          #:vcenter "+"
          (#:column (two num))))
      )))

\relative c' {
  \override Staff.TimeSignature #'stencil = #(compound-time "2" "3" "8")
  \time 5/8
  #(override-auto-beam-setting '(end 1 8 5 8) 1 4)
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```



Véase también

Glosario musical: Sección “polimétrica” in *Glosario Musical*, Sección “compás polimétrico” in *Glosario Musical*, Sección “metro” in *Glosario Musical*.

Referencia de la notación: [Indicación de compás], página 46, [Escalar las duraciones], página 35.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TimeSignature” in *Referencia de Funcionamiento Interno*, Sección “Timing_translator” in *Referencia de Funcionamiento Interno*, Sección “Default_bar_line_engraver” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos
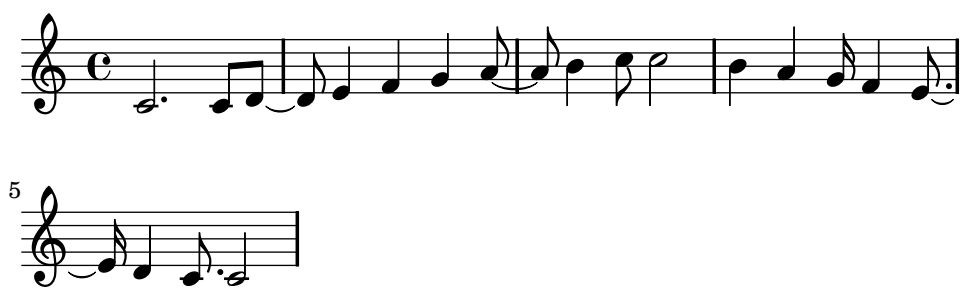
Al usar distintos compases en paralelo, las notas que están en el mismo instante de tiempo se colocan en la misma posición horizontal. Sin embargo, las barras de compás en los distintos pentagramas harán que el espaciado de notas sea menos regular en cada pentagrama individual de lo que sería normal sin las distintas indicaciones de compás.

División automática de las notas

Las notas largas se pueden convertir automáticamente en notas ligadas. Se hace mediante la sustitución del `Note_heads_engraver` por el `Completion_heads_engraver`. En el ejemplo siguiente, las notas que atraviesan la barra de compás se dividen y se unen mediante una ligadura.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
}

{ c2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2 }
```



Este grabador divide todas las notas largas en la barra de compás e inserta ligaduras. Uno de sus usos es depurar partituras complejas: si los compases no están completos, las ligaduras mostrarán exactamente cuánto le falta a cada compás.

Véase también

Glosario musical: Sección “ligadura de unión” in *Glosario Musical*

Manual de aprendizaje: Sección “Engravers explained” in *Manual de Aprendizaje*, Sección “Adding and removing engravers” in *Manual de Aprendizaje*.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “Completion_heads_engraver” in *Referencia de Funcionamiento Interno*, Sección “Forbid_line_break_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

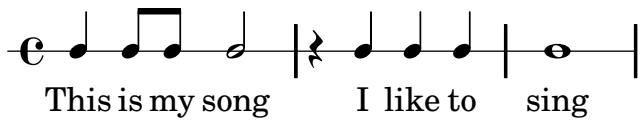
No todas las duraciones (sobre todo las que contienen grupos especiales) se pueden representar exactamente con notas normales y puntillos, pero el grabador `Completion_heads_engraver` nunca insertará grupos especiales.

`Completion_heads_engraver` afecta sólo a las notas; no divide silencios.

Mostrar los ritmos de la melodía

A veces podemos querer mostrar solamente el ritmo de una melodía. Esto se puede hacer con un pentagrama de ritmo. Todas las alturas de las notas se convierten en barras inclinadas, y el pentagrama tiene una sola línea:

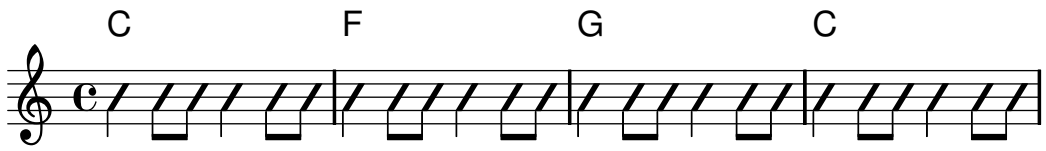
```
<<
  \new RhythmicStaff {
    \new Voice = "misDuraciones" {
      \time 4/4
      c4 e8 f g2
      r4 g g f
      g1
    }
  }
  \new Lyrics {
    \lyricsto "misDuraciones" {
      This is my song
      I like to sing
    }
  }
>>
```



Las tablas de acordes de guitarra ofrecen a menudo los ritmos de rasgueado. Esto se puede hacer con el grabador `Pitch_squash_engraver` y `\improvisationOn`.

```
<<
  \new ChordNames {
    \chordmode {
      c1 f g c
    }
  }

  \new Voice \with {
    \consists Pitch_squash_engraver
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
>>
```



Instrucciones predefinidas

`\improvisationOn`, `\improvisationOff`.

Fragmentos de código seleccionados

Ritmos rasgueados de guitarra

Para la música de guitarra, es posible mostrar los ritmos de rasgueo, además de las notas de la melodía, acordes y diagramas de posiciones.

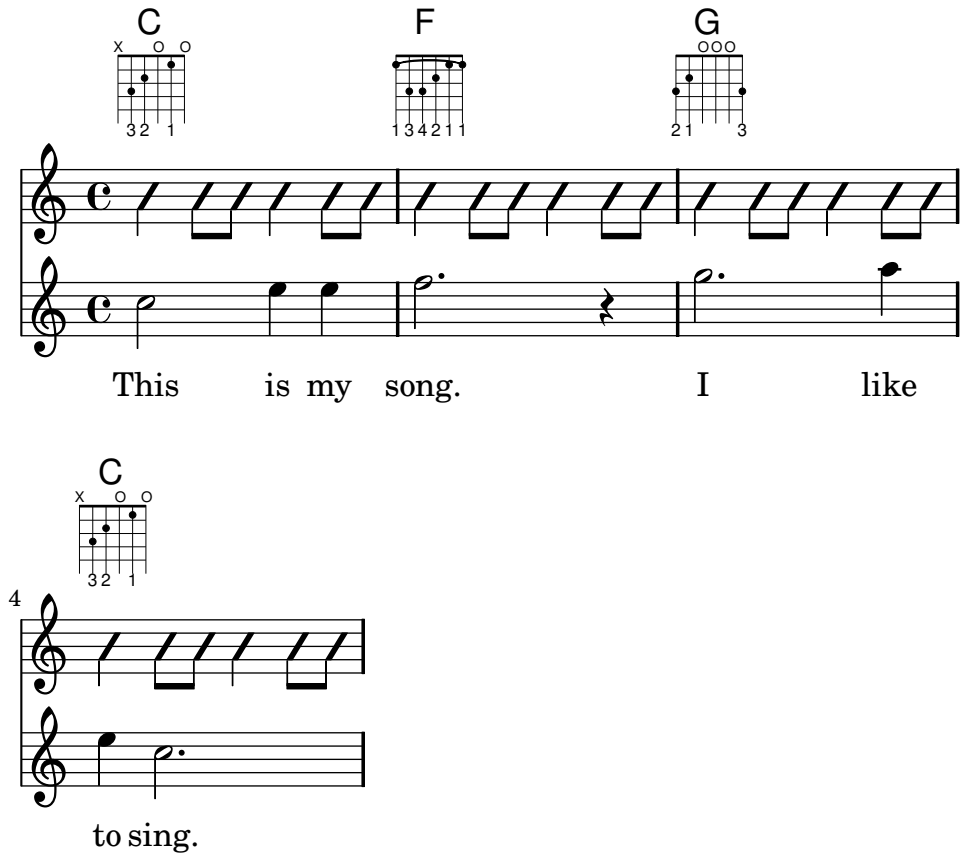
```
\include "predefined-guitar-fretboards.ly"
<<
  \new ChordNames {
    \chordmode {
      c1 f g c
    }
  }

  \new FretBoards {
    \chordmode {
      c1 f g c
    }
  }


  \new Voice \with {
    \consists Pitch_squash_engraver
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }


  \new Voice = "melodia" {
    \relative c'' {
      \improvisationOff
      c2 e4 e4
      f2. r4
      g2. a4
      e4 c2.
    }
  }

  \new Lyrics {
    \lyricsto "melodia" {
      This is my song.
      I like to sing.
    }
  }
>>
```



Véase también

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.
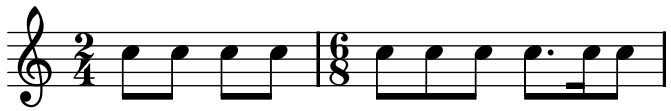
Referencia de funcionamiento interno: Sección “RhythmicStaff” in *Referencia de Funcionamiento Interno*, Sección “Pitch_squash_engraver” in *Referencia de Funcionamiento Interno*.

1.2.4 Barras

Barras automáticas

De manera predeterminada, las barras de corchea se insertan automáticamente:

```
\time 2/4 c8 c c c
\time 6/8 c c c c8. c16 c8
```
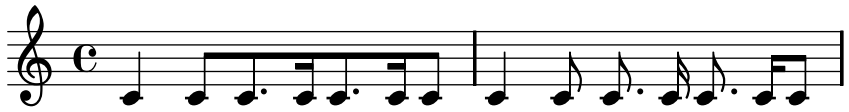


Cuando estas decisiones automáticas no son lo bastante buenas, se pueden escribir los barrados de forma explícita; véase [Barras manuales], página 67. También es posible definir patrones de barrado que difieran de los ajustes por defecto. Las reglas de barrado predeterminadas están definidas en el archivo ‘`scm/auto-beam.scm`’. Si no hay definidas reglas de barrado para una duración dada de la barra en el tipo de compás en uso, el barrado se controla mediante los valores de tres propiedades de contexto, `measureLength`, `beatLength` y `beatGrouping`. Tanto las reglas de barrado como las propiedades de contexto se pueden sobreescribir, véase [Establecer el comportamiento de las barras automáticas], página 58.

Nota: Si se usan barras para indicar los melismas de las canciones, entonces se debe desactivar el barrado automático con `\autoBeamOff` e indicar las barras manualmente.

El barrado automático se puede habilitar o desactivar con las instrucciones `\autoBeamOff` y `\autoBeamOn`:

```
c4 c8 c8. c16 c8. c16 c8
\autoBeamOff
c4 c8 c8. c16 c8.
\autoBeamOn
c16 c8
```



Instrucciones predefinidas

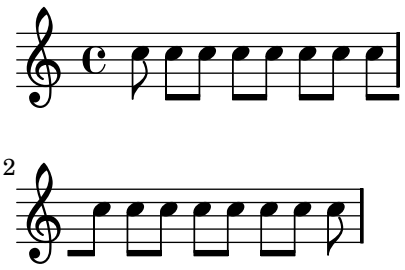
`\autoBeamOff`, `\autoBeamOn`.

Fragmentos de código seleccionados

Beams across line breaks

Normalmente están prohibidos los saltos de línea si las barras atraviesan las líneas divisorias. Se puede cambiar este comportamiento como se muestra aquí:
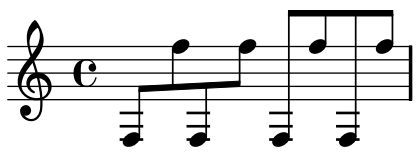
```
\relative c'' {
  \override Beam #'breakable = ##t
  c8 c[ c] c[ c] c[ c] c[ \break
  c8] c[ c] c[ c] c[ c] c
}
```



Cambiar el salto de las barras en ángulo

Se insertan automáticamente barras en ángulo cuando se detecta un intervalo muy grande entre las notas. Se puede hacer un ajuste fino de este comportamiento a través de la propiedad `auto-knee-gap`. Se traza una barra doblada si el salto es mayor que el valor de `auto-knee-gap` más el ancho del objeto barra (que depende de la duración de las notas y de la inclinación de la barra). De forma predeterminada `auto-knee-gap` está establecido a 5.5 espacios de pentagrama.

```
{
  f8 f''8 f8 f''8
  \override Beam #'auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



Véase también

Referencia de la notación: [Barras manuales], página 67, [Establecer el comportamiento de las barras automáticas], página 58.

Archivos instalados: ‘`scm/auto-beam.scm`’.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Beam” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No se pueden usar conjuntamente barras dobladas automáticamente que cruzan pentagramas y pentagramas ocultos. Véase [Ocultar pentagramas], página 139.

Las barras pueden colisionar con las cabezas de las notas y las alteraciones accidentales de las otras voces.

Establecer el comportamiento de las barras automáticas

La colocación de barras automáticas viene determinada por las reglas descritas en [Barras automáticas], página 56. Existen dos formas mutuamente excluyentes de modificar estas reglas. La primera es modificar la agrupación de los pulsos y se aplica a compases poco usuales, es decir, aquellos para los que no ha reglas predefinidas que definen los puntos de final de las barras. El segundo método, modificar la especificación de los puntos finales de las barras, se puede usar para cualquier tipo de compás. Este segundo método **se debe** usar para aquellos compases y combinaciones de duración de barras para los que existen reglas predefinidas de final de barra, a no ser que se hayan revertido. Hay reglas predefinidas para los compases de 3/2, 3/4, 4/4, 2/4, 4/8, 4/16, 6/8, 9/8 y 12/8.

Modificar las agrupaciones de pulsos

Si no hay definida ninguna regla de final de barra para la duración de una barra en particular dentro del tipo de compás en uso, su barrado se controla mediante tres propiedades de contexto: `measureLength`, `beatLength` y `beatGrouping`. Estas propiedades se deben establecer en los contextos `Score`, `Staff` o `Voice` para delimitar el ámbito de su efecto. Los valores predeterminados se establecen al procesarse las instrucciones `\time`, por ello las instrucciones `\set` se deben escribir después de todas las instrucciones `\time`.

Dichas propiedades determinan el barrado de la siguiente forma:

Las barras pueden dar comienzo en cualquier lugar (excepto si ya hay una barra activa). Las barras terminan en un tiempo determinado por los valores de `beatGrouping` y `beatLength`, com se ve a continuación:

- Si `beatGrouping` y `beatLength` con consistentes con `measureLength`, se utiliza `beatGrouping` para determinar los puntos finales de las barras.
- Si `beatGrouping` y `beatLength` no concuerdan con `measureLength`, se usa `beatLength` para determinar los puntos finales de las barras.

Nota: Estas tres propiedades se hacen efectivas para una barra concreta **solamente** si no hay reglas de final de barra predefinidas para esa duración de barra en el tipo de compás en uso, o si todas esas reglas de final de barra se han revertido.

De forma predeterminada, las propiedades the `measureLength` y `beatLength` se derivan del compás establecido por la instrucción `\time`. La longitud `measureLength` está establecida de forma que sea exactamente la misma que la longitud del compás dado por la indicación de compás, y la longitud del pulso `beatLength` tiene un valor dado igual a una parte de compás (1/n, siendo n el denominador del compás).

El valor predeterminado de `beatGrouping` se toma de una tabla que está en '`scm/music-functions.scm`'. Para localizarla, véase Sección "Other sources of information" in *Manual de Aprendizaje*. Define la agrupación de pulsos para los compases de 5/8, 6/8, 8/8, 9/8 y 12/8.

Tanto `measureLength` como `beatLength` son *momentos*, unidades de duración musical. Las cantidades del tipo *moment* se crean por la función de Scheme `ly:make-moment`. Para ver más información sobre esta función, consulte [Gestión del tiempo], página 83.
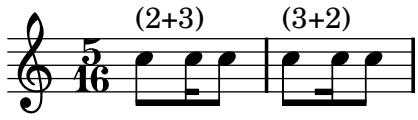
`beatGrouping` es una lista de enteros que dan el número de pulsos en cada grupo.

Fragmentos de código seleccionados

Agrupar los pulsos

Los patrones de barrado se pueden alterar con la propiedad `beatGrouping`:

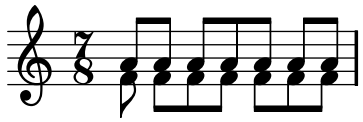
```
\relative c'' {
  \time 5/16
  \set beatGrouping = #'(2 3)
  c8[^"(2+3)" c16 c8]
  \set beatGrouping = #'(3 2)
  c8[^"(3+2)" c16 c8]
}
```



Especificar el contexto con beatGrouping

Mediante la especificación del contexto, el efecto de `beatGrouping` puede limitarse al contexto especificado, y sobreescribirse los valores establecidos en contextos de niveles más altos:

```
\score {
  \new Staff <<
    \time 7/8
    \new Voice {
      \relative c'' {
        \set Staff.beatGrouping = #'(2 3 2)
        a8 a a a a a a
      }
    }
    \new Voice {
      \relative c' {
        \voiceTwo
        \set Voice.beatGrouping = #'(1 3 3)
        f8 f f f f f f
      }
    }
  >>
}
```



Utilización de beatLength y beatGrouping

La propiedad `measureLength` determina dónde se deben insertar líneas divisorias y, con `beatLength` y `beatGrouping`, cómo se deben generar las barras autoomáticas para las duraciones de barra y compases para los que no hay ninguna regla definida para los finales de barra. Este ejemplo muestra distintas dormas de controlar el barrado mediante el establecimiento de estas propiedades. Las explicaciones están en forma de comentarios dentro del código.

```
\relative c'' {
  \time 3/4
  % De forma predeterminada, en 3/4 el barrado se hace en tres grupos
  % de una negra de longitud cada uno
  a16 a a a a a a a a a a

  \time 12/16
  % No hay ningún barrado automático definido para 12/16
  a16 a a a a a a a a a a
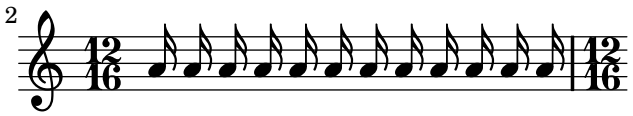
  \time 3/4
  % Cambiar el símbolo de compás, manteniendo el barrado subyacente de 3/4
  \set Score.timeSignatureFraction = #'(12 . 16)
  a16 a a a a a a a a a a

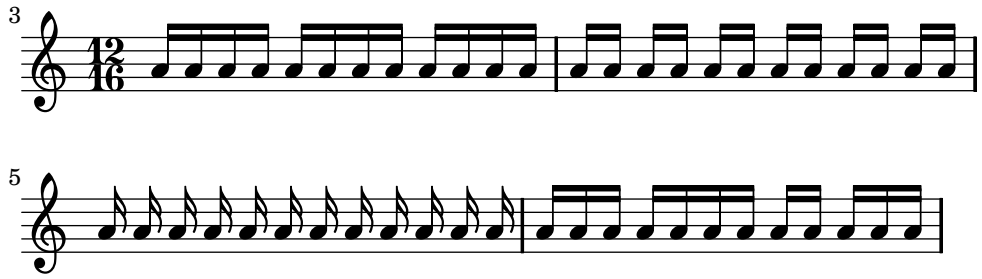
  % El agrupamiento predeterminado de (1 1 1) y beatLength de 1/8 del compás 3/4
  % no son consistentes con una longitud de compás measureLength de 3/4, por ello las barras
  % se agrupan a intervalos de beatLength
  \set Score.beatLength = #(ly:make-moment 1 8)
  a16 a a a a a a a a a a

  % Especificar barras en grupos de (3 3 2 3) semicorcheas
  % 3+3+2+3=11, y 11*1/16<>3/4, así que no es de aplicación beatGrouping
  % y las barras se agrupan a intervalos de beatLength (semicorcheas)
  \set Score.beatLength = #(ly:make-moment 1 16)
  \set Score.beatGrouping = #'(3 3 2 3)
  a16 a a a a a a a a a a

  % Especificat barras en grupos de (3 4 2 3) semicorcheas
  % 3+4+2+3=12, y 12*1/16=3/4, así pues beatGrouping es de aplicación
  \set Score.beatLength = #(ly:make-moment 1 16)
  \set Score.beatGrouping = #'(3 4 2 3)
  a16 a a a a a a a a a a
}
```
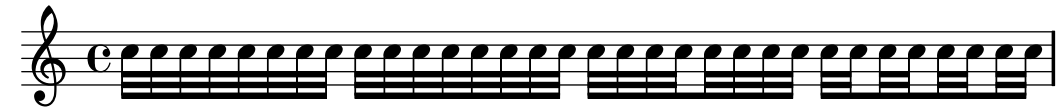






Subdivisión de las barras de semicorchea

Las barras de semicorcheas (o notas más breves) seguidas no se subdividen de forma predeterminada. Esto es: las tres (o más) barras se prolongan, sin dividirse, sobre grupos completos de notas. Este comportamiento se puede modificar para que las barras se subdividan en subgrupos mediante el establecimiento de la propiedad `subdivideBeams`. Cuando está establecida, las diversas barras se subdividen a intervalos definidos por el valor actual de la longitud del pulso `beatLength` reduciendo las barras múltiples a una sola entre los subgrupos. Observe que el valor predeterminado de `beatLength` es de una negra si no está establecido explícitamente. Se debe establecer al valor de una fracción que da la duración el subgrupo de barras utilizando la función `make-moment`, como se muestra aquí:

```
\relative c'' {
  c32[ c c c c c c c]
  \set subdivideBeams = ##t
  c32[ c c c c c c c]

  % Fijar la longitud del subgrupo de barras a una corchea
  \set beatLength = #(ly:make-moment 1 8)
  c32[ c c c c c c c]

  % Fijar la longitud del subgrupo de barras a una semicorchea
  \set beatLength = #(ly:make-moment 1 16)
  c32[ c c c c c c c]
}
```

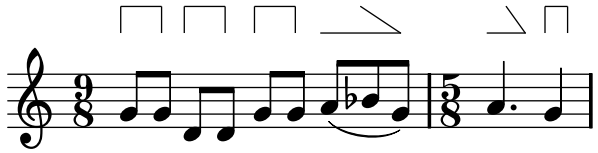


Símbolos de dirección, símbolos de agrupación de compás

Las opciones para agrupar los pulsos de un compás están a nuestra disposición a través de la función de Scheme `set-time-signature`, que acepta tres argumentos: el número de pulsos, la longitud del pulso, y la agrupación interna de los pulsos dentro del compás. Si se incluye el grabador `Measure_grouping_engraver`, la función crea también símbolos de agrupación `MeasureGrouping`. Dichos símbolos facilitan la lectura de la música moderna de cierta complejidad rítmica. En el ejemplo, el compás de 9/8 se subdivide en 2, 2, 2 y 3. Esto se pasa a la función `set-time-signature` como tercer argumento: `'(2 2 2 3)`:

```
\score {
  \relative c'' {
    #(set-time-signature 9 8 '(2 2 2 3))
    #(revert-auto-beam-setting '(end * * 9 8) 3 8)
    #(override-auto-beam-setting '(end 1 8 9 8) 1 4)
    #(override-auto-beam-setting '(end 1 8 9 8) 2 4)
    #(override-auto-beam-setting '(end 1 8 9 8) 3 4)
    g8 g d d g g a( bes g) |
    #(set-time-signature 5 8 '(3 2))
```

```
    a4. g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



Modificar los puntos finales de las barras

En compases de tipo normal, las barras automáticas pueden comenzar en cualquier nota pero solamente pueden terminar en algunas poriciones dentro del compás, concretamente en las duraciones que estén especificadas por las propiedades de `autoBeamSettings`. Estas propiedades consisten en una lista de reglas que especifican dónde pueden terminar las barras. Las reglas predeterminadas de `autoBeamSettings` están definidas en '`scm/auto-beam.scm`'. Para localizarlo, consulte Sección "Other sources of information" in *Manual de Aprendizaje*.

Se debe usar este método para los tipos de compás para los que existen reglas de final de barra definidas de forma predeterminada, a no ser que todas ellas se hayan revertido. También es particularmente apropiado para otros muchos compases si la indicación de compás cambia con frecuencia, o si el barrado debe ser distinto para las distintas duraciones de barra.

Para añadir una regla a la lista, utilice

```
#(override-auto-beam-setting
  '(beam-limit
    beam-numerator  beam-denominator
    time-signature-numerator time-signature-denominator)
  moment-numerator moment-denominator [context])
```

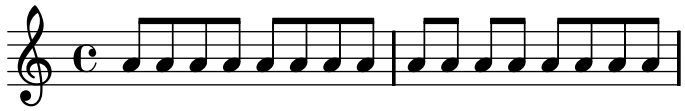
donde:

- `límite-barra` es el tipo de límte de barra automática que se define. Puede ser `begin` o `end`, pero sólo es efectivo `end`.
- `numerador-barra/denominador-barra` es la duración de la barra a la que se quiere aplicar la regla. Se considera que una barra tiene la duración de su nota más breve. Ajuste `numerador-barra` y `denominador-barra` a un asterisco `'*'` para que la regla se aplique a las barras de cualquier duración.
- `numerador-compás/denominador-compás` especifica la indicación de compás a que se aplica esta regla. Si `numerador-compás` y `denominador-compás` tienen el valor de un asterisco `'*'`, esta regla se aplica a cualquier tipo de compás.
- `numerador-momento/denominador-momento` es la posición dentro del compás en que la barra debe terminar.
- `contexto` es opcional, y especifica el contexto en que se debe hacer el cambio. El valor predeterminado es `'Voice`.

  `#(score-override-auto-beam-setting '(A B C D) E F)` es equivalente a `#(override-auto-beam-setting '(A B C D) E F 'Score)`.
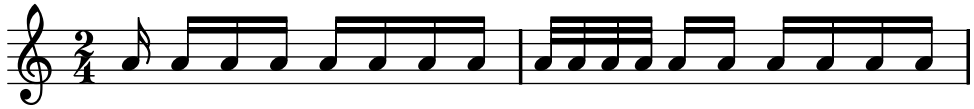
Por ejemplo, si las barras automáticas siempre deben terminar sobre la primera negra, para cualquier tipo de compás o duración de la barra, use
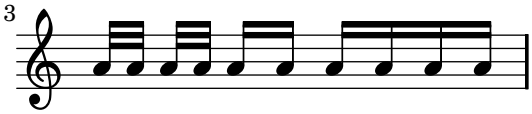
```
a8 a a a a a a a
#(override-auto-beam-setting '(end * * * *) 1 4)
a8 a a a a a a a
```



Podemos forzar los ajustes de las barras para que tengan efecto sólo sobre las barras cuya nota más breve sea de una cierta duración:
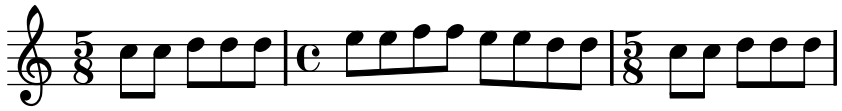
```
\time 2/4
% finalizar barras de semicorchea para todos los compases en el momento 1/16
#(override-auto-beam-setting '(end 1 16 * *) 1 16)
a16 a a a a a a a |
a32 a a a a16 a a a a a |
% finalizar barras de fusa para todos los compases en el momento 1/16
#(override-auto-beam-setting '(end 1 32 * *) 1 16)
a32 a a a a16 a a a a a |
```





Podemos forzar los ajustes de las barras para que tengan efecto solamente en ciertos tipos de compás:

```
\time 5/8
% finalizar barras para cualquier duración en el compás de 5/8, en el momento 2/8
#(override-auto-beam-setting '(end * * 5 8) 2 8)
c8 c d d d
\time 4/4
e8 e f f e e d d
\time 5/8
c8 c d d d
```
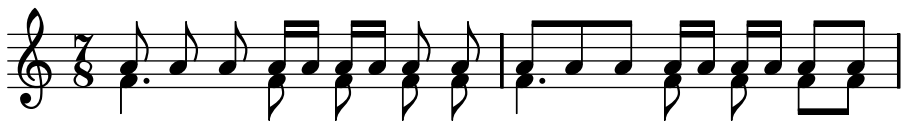


En caso de usar varias voces, se debe especificar el contexto `Staff` si queremos aplicar el barrado a todas las voces del pentagrama:

```
\time 7/8
% ritmo 3-1-1-2
% Contexto sin especificar: no funciona correctamente
#(override-auto-beam-setting '(end * * 7 8) 3 8)
```

```
#(override-auto-beam-setting '(end * * 7 8) 4 8)
#(override-auto-beam-setting '(end * * 7 8) 5 8)
<< {a8 a a a16 a a a a8 a} \\ {f4. f8 f f f} >>

% Funciona correctamente con el contexto especificado
#(override-auto-beam-setting '(end * * 7 8) 3 8 'Staff)
#(override-auto-beam-setting '(end * * 7 8) 4 8 'Staff)
#(override-auto-beam-setting '(end * * 7 8) 5 8 'Staff)
<< {a8 a a a16 a a a a8 a} \\ {f4. f8 f f f} >>
```



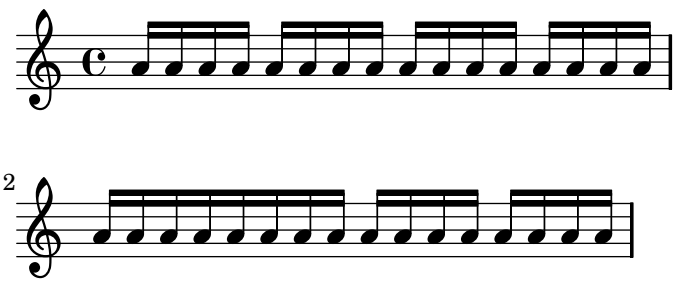
Nota: Si se produce un comportamiento inesperado de las barras, compruebe la existencia de una posible interferencia en los ajustes predeterminados del barrado automático que están en '`scm/auto-beam.scm`', porque los finales de barra definidos aquí aún son válidos además de los suyos.

Debemos revertir cualquier final predeterminado indeseado o conflictivo, para nuestro compás concreto. Podemos eliminar una regla de barrado automático previamente establecida usando

```
#(revert-auto-beam-setting
  '(beam-limit
    beam-numerator  beam-denominator
    time-signature-numerator time-signature-denominator)
  moment-numerator moment-denominator [context])
```

`límite-barra`, `numerador-barra`, `denominador-barra`, `numerador-compás`, `denominador-compás`, `numerador-momento`, `denominador-momento` y `contexto` son los mismos que antes.
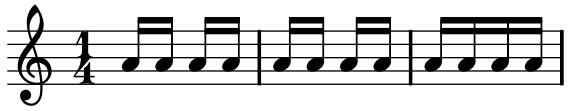
```
\time 4/4
a16 a a a a a a a a a a a a a a a a
% deshacer regla de finalizar barras de semicorchea en compás 4/4 en el momento 1/4
#(revert-auto-beam-setting '(end 1 16 4 4) 1 4)
a16 a a a a a a a a a a a a a a a a
```



La regla de un enunciado `revert-auto-beam-setting` debe corresponderse exactamente con la regla original. Esto es, no se tiene en cuenta ninguna expansión de comodines.

```
\time 1/4
#(override-auto-beam-setting '(end 1 16 1 4) 1 8)
a16 a a a
#(revert-auto-beam-setting '(end 1 16 * *) 1 8) % ¡esto no lo devuelve a su valor anterior!
```

```
a a a a
#(revert-auto-beam-setting '(end 1 16 1 4) 1 8) % esto sí
a a a a
```
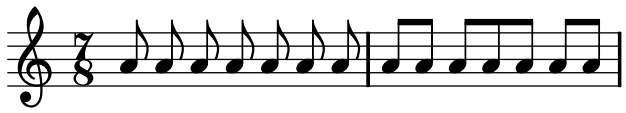


Fragmentos de código seleccionados

Agrupamiento de las barras en el compás de 7/8

No está especificada ninguna agrupación predeterminada automática de las barras para el compás de 7/8, de forma que si se requieren barras automáticas se debe especificar la forma de agrupamiento. Por ejemplo, para agrupar todas las barras en la forma 2-3-2 en el compás de 7/8, especificamos los finales de barra en 2/8 y 5/8:

```
\relative c'' {
  \time 7/8
  % ritmo 2-3-2
  a8 a a a a a a
  #(override-auto-beam-setting '(end * * 7 8) 2 8)
  #(override-auto-beam-setting '(end * * 7 8) 5 8)
  a8 a a a a a a
}
```



Alteración de los finales de barra predeterminados

Para tipografiar las barras agrupadas en la forma `3-4-3-2` en 12/8, en primer lugar tenemos que sobreescribir los finales de barra predeterminados en 12/8, y después preparar los finales de barra nuevos:

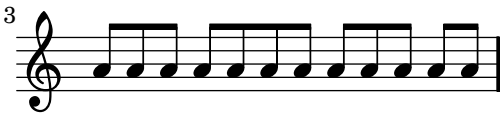
```
\relative c'' {
  \time 12/8

  % Barrado predeterminado
  a8 a a a a a a a a a a a

  % Revertir los valores predeterminados en scm/auto-beam.scm para el 12/8
  #(revert-auto-beam-setting '(end * * 12 8) 3 8)
  #(revert-auto-beam-setting '(end * * 12 8) 3 4)
  #(revert-auto-beam-setting '(end * * 12 8) 9 8)
  a8 a a a a a a a a a a a

  % establecer valores nuevos para finales de barra
  #(override-auto-beam-setting '(end * * 12 8) 3 8)
  #(override-auto-beam-setting '(end * * 12 8) 7 8)
  #(override-auto-beam-setting '(end * * 12 8) 10 8)
  a8 a a a a a a a a a a a
}
```
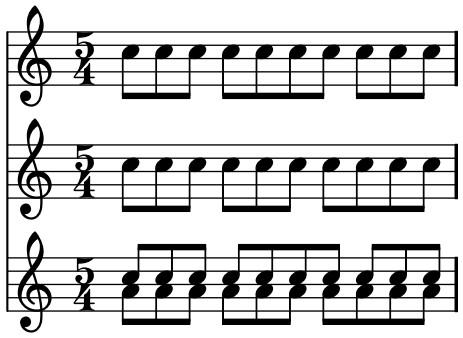




Finales de barra en el contexto Score

Las reglas de final de barra especificadas en el contexto `Score` se aplican a todos los pentagramas, pero se pueden modificar tanto en los niveles de `Staff` como de `Voice`:

```
\relative c'' {
  \time 5/4
  % Establecer el barrado predeterminado para todos los pentagramas
  #(score-override-auto-beam-setting '(end * * 5 4) 3 8)
  #(score-override-auto-beam-setting '(end * * 5 4) 7 8)
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modificar el barrado sólo para este pentagrama
      #(override-auto-beam-setting '(end * * 5 4) 6 8 'Staff)
      #(revert-auto-beam-setting '(end * * 5 4) 7 8 'Staff)
      c8 c c c c c c c c c
    }
    \new Staff {
      % Heredar el barrado del contexto Score
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
        % Modificar el barrado sólo para esta voz
        \new Voice {
          \voiceTwo
          #(override-auto-beam-setting '(end * * 5 4) 6 8)
          #(revert-auto-beam-setting '(end * * 5 4) 7 8)
          a8 a a a a a a a a a
        }
      >>
    }
  >>
}
```



Instrucciones predefinidas

`\autoBeamOff`, `\autoBeamOn`.

Advertencias y problemas conocidos

Si una partitura termina mientras una barra automática no ha terminado aún todavía está aceptando notas, esta última barra no se imprimirá en absoluto. Lo mismo sirve para las voces polifónicas, introducidas con `<< ... \\ ... >>`. Si una voz polifónica termina mientras una barra automática está aún aceptando notas, no se imprime.
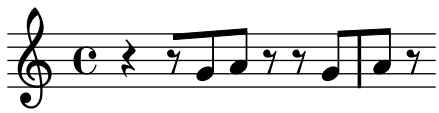
Véase también

Fragmentos de código: Sección "Rhythms" in *Fragmentos de código*.

Barras manuales
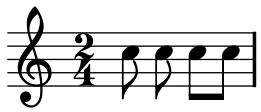
En ciertos casos puede ser preciso sobreescribir el algoritmo de barrado automático. Por ejemplo, el barrador automático no escribe barras por encima de los silencios o las líneas divisorias, y en las partituras corales el barrado se ajusta con frecuencia para que siga la medida de la letra en vez de la de las notas. Tales barras se especifican manualmente marcando los puntos de comienzo y final con `[` y `]`

```
{
  r4 r8[ g' a r8] r8 g[ | a] r8
}
```
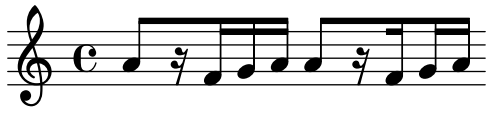


Se pueden marcar notas individuales con `\noBeam` para evitar que resulten unidas por una barra:

```
\time 2/4 c8 c\noBeam c c
```



Se puede conseguir un control incluso más estritcto sobre las barras estableciendo las propiedades `stemLeftBeamCount` y `stemRightBeamCount`. Especifican el número de barras que se dibujarán en los lados izquierdo y derecho, respectivamente, de la nota siguiente. Si cualquiera de estas dos propiedades está ajustada a un valor, dicho valor se usará una sola vez, y luego se borrará. En este ejemplo, el último Fa se imprime con sólo una barra en el lado izquierdo, es decir, la barra de corchea del grupo como un todo.

```
a8[ r16 f g a]
a8[ r16
\set stemLeftBeamCount = #2
\set stemRightBeamCount = #1
f
\set stemLeftBeamCount = #1
g a]
```



Fragmentos de código seleccionados

Corchetes rectos y extremos de barra sueltos

Son posibles tanto los corchetes rectos sobre notas sueltas como extremos de barra sueltos en figuras unidas, con una combinación de `stemLeftBeamCount`, `stemRightBeamCount` e indicadores de barra `[ ]` emparejados.

Para corchetes rectos que apunten a la derecha sobre notas sueltas, use indicadores de barra emparejados `[ ]` y establezca `stemLeftBeamCount` a cero (véase el ejemplo 1).

Para corchetes rectos que apunten a la izquierda, establezca en su lugar `stemRightBeamCount` (ejemplo 2).

Para extremos sueltos que apunten a la derecha al final de un conjunto de notas unidas, establezca `stemRightBeamCount` a un valor positivo. Y para extremos sueltos que apunten a la izquierda al principio de un conjunto de notas unidas, establezca `stemLeftBeamCount` en su lugar (ejemplo 3).

A veces, para una nota suelta rodeada de silencios tiene sentido que lleve los dos extremos sueltos del corchete plano, apuntando a derecha e izquierda. Hágalo solamente con indicadores de barra emparejados `[ ]` (ejemplo 4).

(Observe que `\set stemLeftBeamCount` siempre equivale a `\once \set`. En otras palabras, los ajustes de la cantidad de barras no se recuerdan, y por ello el par de corchetes planos aplicados a la nota Do semicorchea `c'16 [ ]` del último ejemplo no tiene nada que ver con el `\set` de dos notas por detrás.)

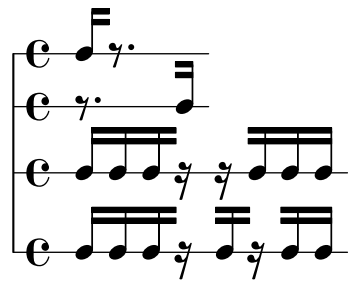
```
\score {
<<
% Ejemplo 1
\new RhythmicStaff {
  \set stemLeftBeamCount = #0
  c'16 [ ]
  r8.
}

% Ejemplo 2
\new RhythmicStaff {
  r8.
  \set stemRightBeamCount = #0
  c'16 [ ]
}

% Ejemplo 3
```

```
\new RhythmicStaff {
  c'16
  c'16
  \set stemRightBeamCount = #2
  c'16
  r16
  r16
  \set stemLeftBeamCount = #2
  c'16
  c'16
  c'16
}

% Ejemplo 4
\new RhythmicStaff {
   c'16
   c'16
   \set stemRightBeamCount = #2
   c'16
   r16
   c'16 [ ]
   r16
   \set stemLeftBeamCount = #2
   c'16
   c'16
}
>>
}
```



Barras progresivas
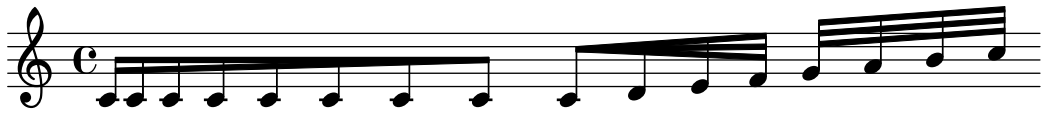
Las barras progresivas se usan para indicar que un pequeño grupo de notas se debe tocar a una velocidad creciente (o decreciente), sin cambiar el tempo general de la pieza. El ámbito de la barra progresiva se debe indicar manualmente usando `[` y `]`, el efecto de ángulo de la barra se inicia especificando una dirección en la propiedad `grow-direction` del objeto `Beam`.

Si queremos que la colocación de las notas y el sonido de la salida MIDI refleje el ritardando o accelerando indicado por la barra progresiva, las notas se deben agrupar como una expresión musical delimitada por llaves y precedida de una instrucción `featheredDurations` que especifica la razón entre las duraciones de la primera y la última notas dentro del grupo.

Los corchetes rectos muestran el ábito de la barra y las llaves muestran qué notas han de modificar sus duraciones. Normalmente delimitarían el mismo grupo de notas, pero no es un requisito: las dos instrucciones son independientes.

En el ejemplo siguiente las ocho semicorcheas ocupan exactamente el mismo tiempo que una blanca, pero la primera nota tiene la mitar de duración que la última, con las notas intermedias alargándose gradualmente. Las cuatro primeras fusas se aceleran gradualmente, mientras que las últimas cuatro fusas están a un tempo constante.

```
\override Beam #'grow-direction = #LEFT
\featherDurations #(ly:make-moment 2 1)
{ c16[ c c c c c c c] }
\override Beam #'grow-direction = #RIGHT
\featherDurations #(ly:make-moment 2 3)
{ c32[ d e f] }
% volver a usar barras continuas
\override Beam #'grow-direction = #'()
{ g32[ a b c] }
```



El espaciado en la salida impresa representa las duraciones de las notas de una forma sólo aproximada, pero la salida MIDI es exacta.

Advertencias y problemas conocidos

La instrucción `\featherDurations` sólo funciona con fragmentos musicales muy breves, y cuando los números de la fracción son pequeños.

Véase también

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.
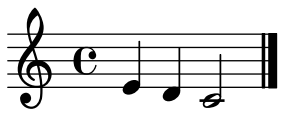
1.2.5 Barras de compás

Barras de compás

Las líneas divisorias delimitan a los compases pero también se pueden usar para indicar las repeticiones. Normalmente, las líneas divisorias normales se insertan de manera automática en la salida impresa en lugares que están basados en el compás actual.

Las barras de compás sencillas insertadas automáticamente se pueden cambiar por otros tipos con la instrucción `\bar`. Por ejemplo, se suele poner una doble barra de cierre al final de la pieza:

```
e4 d c2 \bar "|."
```



No deja de ser válida la última nota de un compás si no termina sobre la línea divisoria automática: se supone que la nota se prolonga sobre el compás siguiente. Pero una secuencia larga de dichos compases prolongados puede hacer que la música aparezca comprimida o incluso que se salga de la página. Esto es a causa de que los saltos de línea automáticos solamente se producen al final de compases completos, es decir, cuando todas las notas han finalizado antes de que el compás termine.

Nota: Una duración incorrecta puede hacer que se trate de evitar la producción de saltos de línea, dando como resultado una línea de música con una compresión exagerada o música que se sale de la página.

También se permiten saltos de línea en las barras insertadas manualmente incluso dentro de compases incompletos. Para permitir un salto de línea donde no hay ninguna línea divisoria visible, utilice

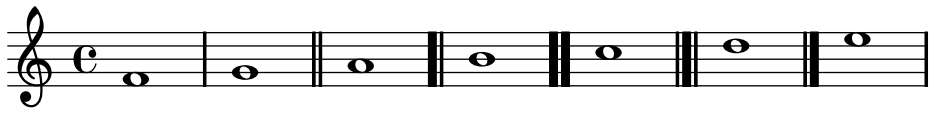
```
\bar ""
```

De esta manera se insertará una barra invisible de compás y se hará posible el salto de línea en este punto, sin forzarlo. No se incrementa el contador de los números de compás. Para forzar un salto de línea, consulte Sección 4.3.1 [Saltos de línea], página 352.

Este y otras líneas divisorias especiales se pueden insertar manualmente en cualquier punto. Cuando coinciden con el final de un compás, sustituyen a la línea divisoria simple que se habría insertado automáticamente en dicho lugar. Cuando no coinciden con el final de un compás, se inserta la línea especificada en ese punto en la salida impresa. Dichas inserciones no afectan al cálculo y colocación de las líneas divisorias automáticas subsiguientes.

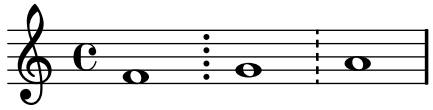
Están disponibles para su inserción manual la línea divisoria simple y cinco tipos de doble barra:

```
f1 \bar "|" g \bar "||" a \bar ".|" b \bar ".|." c \bar "|.|" d \bar "|." e
```



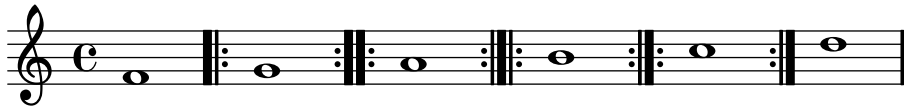
así como la barra de puntos y la discontinua:

```
f1 \bar ":" g \bar "dashed" a
```



y cinco tipos de barra de repetición:
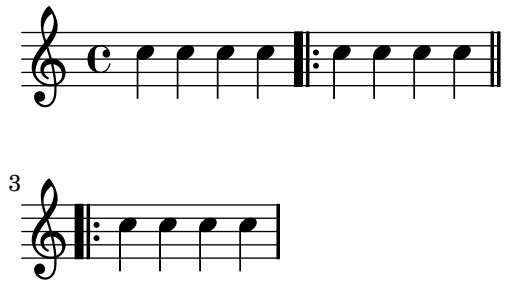
```
f1 \bar "|:" g \bar ":|:" a \bar ":|.|:" b \bar ":|.:" c \bar ":|" d
```



Aunque se pueden insertar manualmente barras de compás con significado de repeticiones, no se reconocen como repeticiones por parte de LilyPond. Las secciones repetidas se introducen mejor utilizando las diversas instrucciones de repetición (véase Sección 1.4 [Repeticiones], página 101), que imprimen automáticamente las barras correspondientes.
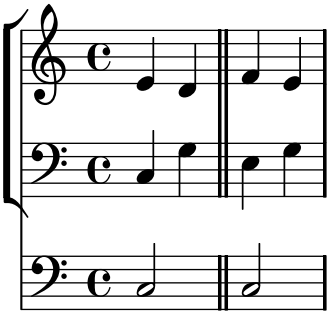
Además se puede especificar `"||:"`, que equivale a `"|:"` excepto en los saltos de línea, en que produce una doble barra al final de la línea y una repetición izquierda (de comienzo) al principio de la línea siguiente.

```
\override Score.RehearsalMark #'padding = #3
c c c c
\bar "||:"
c c c c \break
\bar "||:"
c c c c
```



En las partituras con muchos pentagramas, una instrucción `\bar` en uno de ellos se aplica automáticamente a todos los demás. Las líneas resultantes se conectan entre los distintos pentagramas de un `StaffGroup`, `PianoStaff` o `GrandStaff`.

```
<<
  \new StaffGroup <<
    \new Staff {
      e'4 d'
      \bar "||"
      f' e'
    }
    \new Staff { \clef bass c4 g e g }
  >>
  \new Staff { \clef bass c2 c2 }
>>
```



Fragmentos de código seleccionados

La instrucción `\bar` *tipo_de_barra* es una forma corta de hacer `\set Timing.whichBar =` *tipo_de_barra*. Cuando `whichBar` se establece con el valor de una cadena, se crea una línea divisoria de ese tipo.

El tipo de barra predeterminado que se usa para las líneas divisorias insertadas automáticamente es `"|"`. Se puede cambiar en cualquier momento con `\set Timing.defaultBarType =` *bartype*.

Véase también

Referencia de la notación: Sección 4.3.1 [Saltos de línea], página 352, Sección 1.4 [Repeticiones], página 101, [Agrupar pentagramas], página 127.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

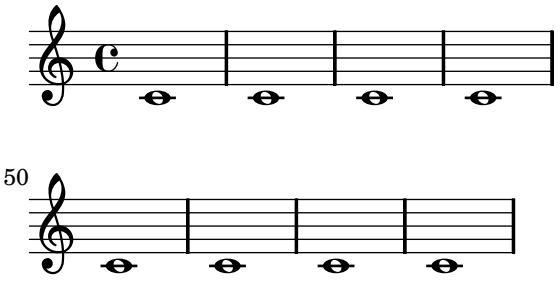
Referencia de funcionamiento interno:

Sección “BarLine” in *Referencia de Funcionamiento Interno* (creada al nivel de Sección “Staff” in *Referencia de Funcionamiento Interno* (pentagrama)), Sección “SpanBar” in *Referencia de Funcionamiento Interno* (a través de los pentagramas), Sección “Timing_translator” in *Referencia de Funcionamiento Interno* (para las propiedades del contador de tiempo Timing).

Numeración de compases

Por defecto, los números de compás se imprimen al principio de la línea, excepto la primera. El número propiamente dicho se almacena en la propiedad `currentBarNumber`, que normalmente se actualiza automáticamente para cada compás. También se puede establecer manualmente:

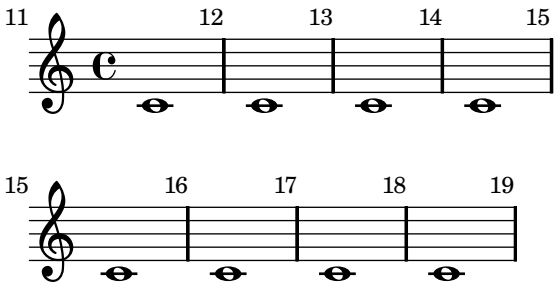
```
c1 c c c
\break
\set Score.currentBarNumber = #50
c1 c c c
```



Fragmentos de código seleccionados

Los números de compás se pueden tipografiar a intervalos regulares en vez de al principio de cada línea. Para hacerlo se debe sobreescribir el comportamiento predeterminado de forma que se puedan imprimir en otros lugares. Esto viene controlado por la propiedad `break-visibility` de `BarNumber`. Toma tres valores que se pueden fijar al valor `#t` o `#f` para especificar si el número de compás correspondiente es visible o no. El orden de los tres valores es `visible al final de la línea`, `visible en mitad de la línea`, `visible al principio de la línea`. En el ejemplo siguiente se imprimen los números de compás en todos los sitios posibles:

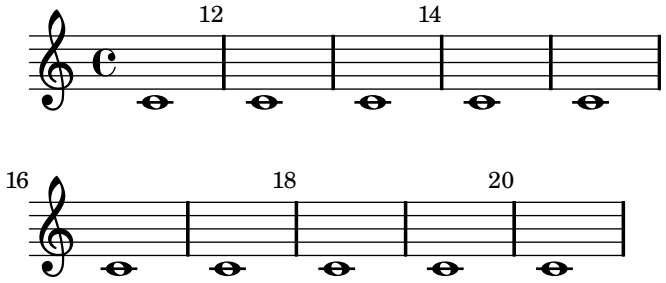
```
\override Score.BarNumber #'break-visibility = #'#(#t #t #t)
\set Score.currentBarNumber = #11
\bar ""  % Habilitar la impresión del número del primer compás
c1 c c c
\break
c c c c
```



y aquí los números de compás se imprimen cada dos compases excepto al final de la línea:

```
\override Score.BarNumber #'break-visibility = #'#(#f #t #t)
\set Score.currentBarNumber = #11
\bar ""  % Habilitar la impresión del número del primer compás
% Imprimir el número de compás cada dos compases
\set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
c1 c c c c
\break
```

```
c c c c c
```



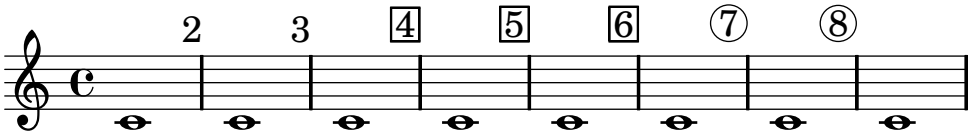
Se puede cambiar el tamaño del número de compás. Esto queda ilustrado en el siguiente ejemplo, que también muestra cómo encerrar los números de compás en rectángulos y circunferencias, y ofrece una forma alternativa de especificar `#(#f #t #t)` para `break-visibility`.

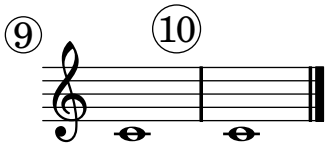
```
% No poner números de compás al final de la línea, pero sí en otros lugares
\override Score.BarNumber #'break-visibility
  = #end-of-line-invisible

% Aumentar en 2 el tamaño del número de compás
\override Score.BarNumber #'font-size = #2
\repeat unfold 3 { c1 } \bar "|"

% Pintar una caja rodeando al siguiente(s) número de compás
\override Score.BarNumber  #'stencil
  = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
\repeat unfold 3 { c1 } \bar "|"

% Rodear con un círculo el siguiente(s) número de compás
\override Score.BarNumber  #'stencil
  = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
\repeat unfold 4 { c1 } \bar "|."
```

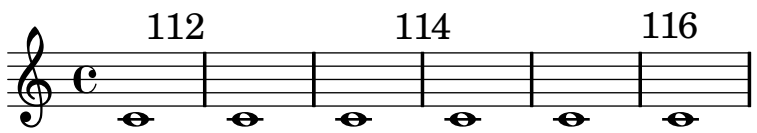




Los números de compás están alineados de forma predeterminada por la izquierda a su objeto padre. Esto suele corresponder al límite izquierdo de una línea o, si los números se imprimen dentro de la línea, la divisoria izquierda del compás. Los números también se pueden colocar directamente sobre la línea divisoria o alinearse por la derecha con ella:

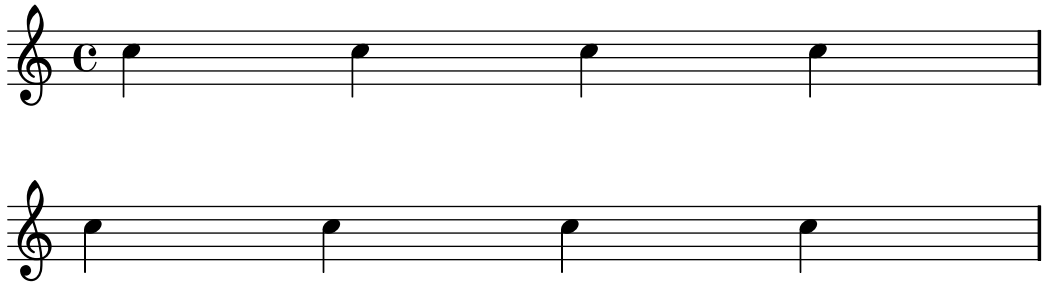
```
\set Score.currentBarNumber = #111
\override Score.BarNumber #'break-visibility = #'#(#t #t #t)
% Aumentar en 2 el tamaño del número de compás
\override Score.BarNumber #'font-size = #2
% Imprimir el número de compás cada dos compases
```

```
\set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
c1 c1
% Alinear números de compás al centro
\override Score.BarNumber #'self-alignment-X = #0
c1 c1
% Alinear números de compás a la derecha
\override Score.BarNumber #'self-alignment-X = #-1
c1 c1
```



Los números de compás se pueden suprimir completamente quitando el grabador `Bar_number_engraver` de números de compás del contexto de partitura `Score`.

```
\layout {
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}
\relative c''{
  c4 c c c \break
  c4 c c c
}
```



Véase también

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

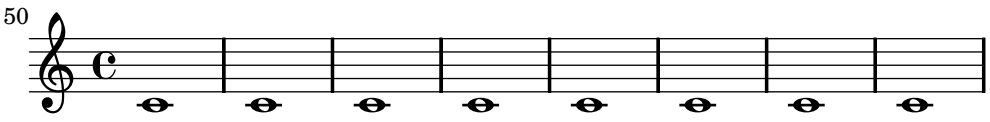
Referencia de funcionamiento interno: Sección “BarNumber” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los números de compás pueden colisionar con el corchete Sección “StaffGroup” in *Referencia de Funcionamiento Interno*, si hay uno en la parte de arriba. Para solucionarlo, se puede usar la propiedad de relleno `padding` de Sección “BarNumber” in *Referencia de Funcionamiento Interno* para colocar el número correctamente.

Los números de compás solamente se pueden imprimir en las líneas divisorias; para imprimir un número de compás al principio de la pieza, se debe insertar allí una barra de compás vacía, y se debe poner un valor distinto a `1` en `currentBarNumber`:

```
\set Score.currentBarNumber = #50
\bar ""
c1 c c c
c1 c c c
\break
```



Comprobación de compás y de número de compás

Las comprobaciones de compás ayudan a detectar errores en las duraciones. Una comprobación de compás se escribe usando el símbolo de la barra vertical, `|`, en cualquier lugar donde se espera que caiga una línea divisoria. Si se encuentran líneas de comprobación de compás en otros lugares, se imprime una lista de advertencias en el archivo log de registro, mostrando los números de línea y columna en que han fallado las comprobaciones de compás. En el siguiente ejemplo, la segunda comprobación de compás avisará de un error

```
\time 3/4 c2 e4 | g2 |
```

Las comprobaciones de compás también se pueden usar en la letra de las canciones, por ejemplo
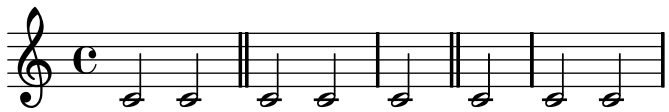
```
\lyricmode {
  \time 2/4
  Bri -- lla | Bri -- lla |
}
```

Las duraciones incorrectas a menudo desbaratan la partitura completamente, sobre todo si la música es polifónica, por ello el mejor sitio para empezar a corregir la entrada es hacer un barrido en busca de comprobaciones de compás fallidas y duraciones incorrectas.

Si varias comprobaciones de compás corresponden al mismo intervalo musical, sólo se imprime el primer mensaje de advertencia. Esto permite que la advertencia se centre en la fuente del error de cuenta de tiempo.

También es posible redefinir la acción que se lleva a cabo cuando se encuentra un símbolo de comprobación de compás o «símbolo de tubería», `|`, en el código de entrada, de forma que haga algo distinto a una comprobación de compás. Se hace asignando una expresión musical a `pipeSymbol` («símbolo de tubería»). En el ejemplo siguiente se establece `|` de forma que inserte una doble línea divisoria cuando aparece en el código de entrada, en vez de comprobar el final de un compás.

```
simboloTubo = \bar "||"
{
  c'2 c'2 |
  c'2 c'2
  c'2 | c'2
  c'2 c'2
}
```



Al copiar piezas musicales grandes puede servir de ayuda comprobar que el número de compás de LilyPond corresponde al original desde el que está copiando. Esto se puede comprobar con `\barNumberCheck`, por ejemplo:

```
\barNumberCheck #123
```

imprimirá una advertencia si el `currentBarNumber` (número del compás actual) no es 123 cuando se procesa.

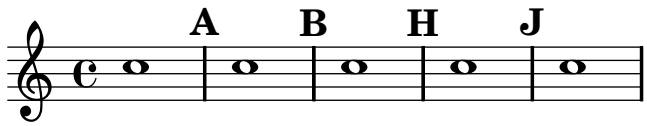
Véase también

Fragmentos de código: Sección "Rhythms" in *Fragmentos de código*.

Llamadas de ensayo

Para imprimir una letra de ensayo, utilice la orden `\mark`

```
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
```



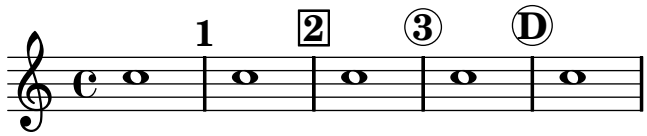
La letra 'I' se salta de acuerdo con las tradiciones de grabado. Si quiere incluir la letra 'I', use

```
\set Score.markFormatter = #format-mark-alphabet
```

La letra de ensayo se incrementa automáticamente si usa `\mark \default`, pero también puede utilizar un número entero como argumento para establecer la indicación manualmente. El valor que se utilizará se almacena dentro de la propiedad `rehearsalMark`.

El estilo se define por medio de la propiedad `markFormatter`. Es una función que toma como argumentos la marca en curso (un número entero) y el contexto actual. Debe devolver un objeto de marcado. En el ejemplo siguiente, se establece `markFormatter` con el valor de un procedimiento enlatado. Después de algunos compases, se establece con el valor de una función que produce un número encerrado en una caja.

```
\set Score.markFormatter = #format-mark-numbers
c1 \mark \default
c1 \mark \default
\set Score.markFormatter = #format-mark-box-numbers
c1 \mark \default
\set Score.markFormatter = #format-mark-circle-numbers
c1 \mark \default
\set Score.markFormatter = #format-mark-circle-letters
c1
```



El archivo 'scm/translation-functions.scm' contiene las definiciones de `format-mark-numbers` (el formato por omisión), `format-mark-box-numbers`, `format-mark-letters` y `format-mark-box-letters`. Se pueden usar éstos como inspiración para otras funciones de formateo.

Podemos utilizar `format-mark-barnumbers`, `format-mark-box-barnumbers` y `format-mark-circle-barnumbers` para obtener números de compás en lugar de números o letras secuenciales.

Se pueden especificar otros estilos de letra de ensayo de forma manual:

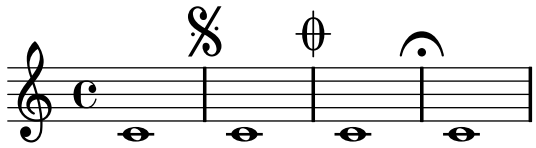
```
\mark "A1"
```

`Score.markFormatter` no afecta a las marcas que se especifican de esta forma. Sin embargo, es posible aplicar un elemento `\markup` a la cadena.

```
\mark \markup{ \box A1 }
```

Los glifos musicales (como el segno) se pueden imprimir dentro de un elemento `\mark`

```
c1 \mark \markup { \musicglyph #"scripts.segno" }
c1 \mark \markup { \musicglyph #"scripts.coda" }
c1 \mark \markup { \musicglyph #"scripts.ufermata" }
c1
```



Consulte Sección B.6 [La tipografía Feta], página 461 para ver una lista de los símbolos que se pueden imprimir con `\musicglyph`.

Para ver formas comunes de trucar la colocación de las letras de ensayo, consulte [Indicaciones de texto], página 167.

Véase también

Referencia de la notación: Sección B.6 [La tipografía Feta], página 461, Sección 1.8.2 [Formatear el texto], página 173.

Archivos de inicio: '`scm/translation-functions.scm`' contiene la definición de `format-mark-numbers` y de `format-mark-letters`. Se pueden usar como inspiración para escribir otras funciones de formateo.

Fragmentos de código: Sección "Rhythms" in *Fragmentos de código*.
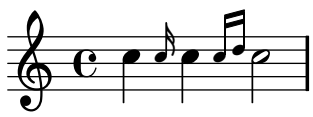
Referencia de funcionamiento interno: Sección "RehearsalMark" in *Referencia de Funcionamiento Interno*.

1.2.6 Asuntos rítmicos especiales

Notas de adorno
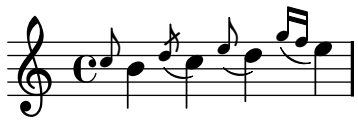
Los mordentes y notas de adorno son ornamentos escritos. Se imprimen en un tipo más pequeño y no ocupan ningún tiempo lógico en el compás.

```
c4 \grace c16 c4
\grace { c16[ d16] } c2
```
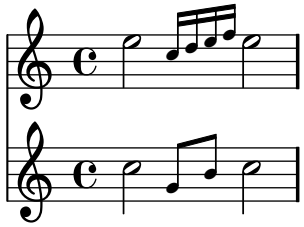


Lilypond también contempla dos tipos especiales de notas de adorno, la *acciaccatura* o mordente de una nota (una nota de adorno sin medida que se indica mediante una pequeña nota ligada y con el corchete tachado) y la *appoggiatura*, que toma una fracción fija de la nota principal, y se denota como una nota pequeña, ligada, sin tachar.

```
\grace c8 b4
\acciaccatura d8 c4
\appoggiatura e8 d4
\acciaccatura { g16[ f] } e4
```
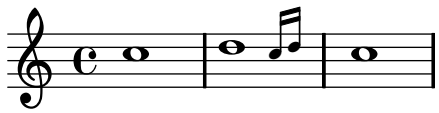


La colocación de notas de adorno se sincroniza entre los distintos pentagramas. En el siguiente ejemplo, hay dos semicorcheas de adorno por cada corchea de adorno:

```
<< \new Staff { e2 \grace { c16[ d e f] } e2 }
   \new Staff { c2 \grace { g8[ b] } c2 } >>
```
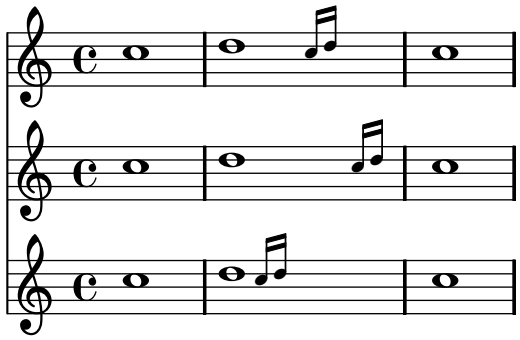


Si queremos terminar una nota con un adorno, usamos la instrucción `\afterGrace`. Toma dos argumentos: la nota principal, y las notas de adorno que siguen a la nota principal.

```
c1 \afterGrace d1 { c16[ d] } c1
```
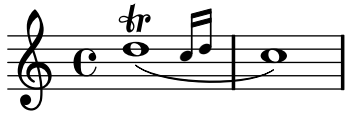


Esto pone las notas de adorno después de un espacio que dura 3/4 de la longitud de la nota principal. La fracción predeterminada de 3/4 se puede cambiar estableciendo `afterGraceFraction`. El ejemplo siguiente muestra el resultado de establecer el espacio en su valor predeterminado, en 15/16, y por último en 1/2 de la nota principal.

```
<<
  \new Staff {
    c1 \afterGrace d1 { c16[ d] } c1
  }
  \new Staff {
    #(define afterGraceFraction (cons 15 16))
    c1 \afterGrace d1 { c16[ d] } c1
  }
  \new Staff {
    #(define afterGraceFraction (cons 1 2))
    c1 \afterGrace d1 { c16[ d] } c1
  }
>>
```



El espacio entre la nota principal y la de adorno también se puede especificar usando espaciadores. El ejemplo siguiente sitúa la nota de adorno después de un espacio que dura 7/8 de la nota principal.

```
\new Voice {
  << { d1^\trill_( }
     { s2 s4. \grace { c16[ d] } } >>
  c1)
}
```



Una expresión musical `\grace` introduce ajustes de tipografía especiales, por ejemplo para producir un tipo más pequeño y para fijar las direcciones. Por ello, cuando se introducen trucos para la presentación, deben ir dentro de la expresión de adorno. Las sobreescrituras se deben también revertir dentro de la expresión de adorno. Aquí, la dirección predeterminada de la plica de la nota de adorno se sobreescribe y luego se revierte.
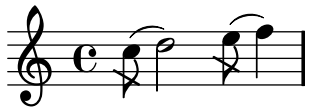
```
\new Voice {
  \acciaccatura {
    \stemDown
    f16->
    \stemNeutral
  }
  g4 e c2
}
```



Fragmentos de código seleccionados
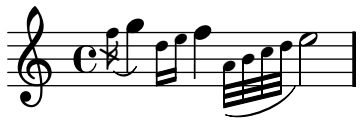
El trazo inclinado que atraviesa al corchete de las *acciaccatura*s se puede aplicar en otras situaciones:

```
\relative c'' {
  \override Stem #'stroke-style = #"grace"
  c8( d2) e8( f4)
}
```



La disposición de las expresiones de adorno se pueden cambiar a lo largo de la música usando la función `add-grace-property`. El siguiente ejemplo suprime la definición de la dirección de la `Stem` (plica) para este adorno, y así las plicas no siempre apuntan hacia arriba.

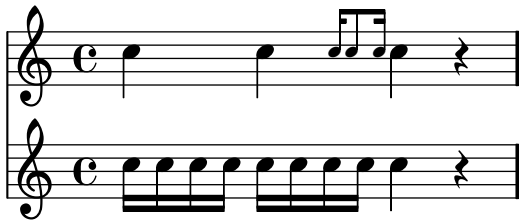
```
\relative c'' {
  \new Staff {
    #(add-grace-property 'Voice 'Stem 'direction ly:stem::calc-direction)
    #(remove-grace-property 'Voice 'Stem 'direction)
    \new Voice {
       \acciaccatura { f16 } g4
       \grace { d16[ e] } f4
       \appoggiatura { a,32[ b c d] } e2
    }
  }
}
```



Otra opción es modificar las variables `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic` y `stopAppoggiaturaMusic`. Sus valores predeterminados están en el archivo '`ly/grace-init.ly`'. Al redefinirlos se pueden obtener otros efectos.

Se puede forzar la alineación de las notas de adorno con las notas normales de los otros pentagramas:

```
\relative c'' {
  <<
    \override Score.SpacingSpanner #'strict-grace-spacing = ##t
    \new Staff {
      c4
      \afterGrace c4 { c16[ c8 c16] }
      c4 r
    }
    \new Staff {
      c16 c c c c c c c c4 r
    }
  >>
}
```



Véase también

Glosario musical: Sección "notas de adorno" in *Glosario Musical*, Sección "acciaccatura" in *Glosario Musical*, Sección "appoggiatura" in *Glosario Musical*.

Archivos de inicio: '`ly/grace-init.ly`'.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.
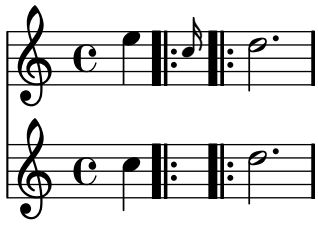
Referencia de funcionamiento interno: Sección “GraceMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Una *acciaccatura* de varias notas con una barra se imprime sin tachar, y tiene exactamente la misma apariencia que una *appoggiatura* de varias notas con barra.
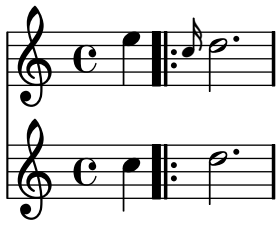
La sincronización de las notas de adorno también puede acarrear sorpresas. La notación de pentagramas, como armaduras, líneas divisorias, etc., se sincronizan también. Ponga cuidado cuando mezcle pentagramas con adornos y sin adornos, por ejemplo

```
<< \new Staff { e4 \bar "|:" \grace c16 d2. }
   \new Staff { c4 \bar "|:" d2. } >>
```



Esto se puede remediar insertando desplazamientos de adorno de las duraciones correspondientes en los otros pentagramas. Para el ejemplo anterior

```
<< \new Staff { e4 \bar "|:" \grace c16 d2. }
   \new Staff { c4 \bar "|:" \grace s16 d2. } >>
```



Las secciones de adorno sólo se deben usar dentro de expresiones de música secuenciales. No están contemplados el anidado ni la yuxtaposición de secciones de adorno, y podría producir caídas u otros errores.

Alinear con una cadenza

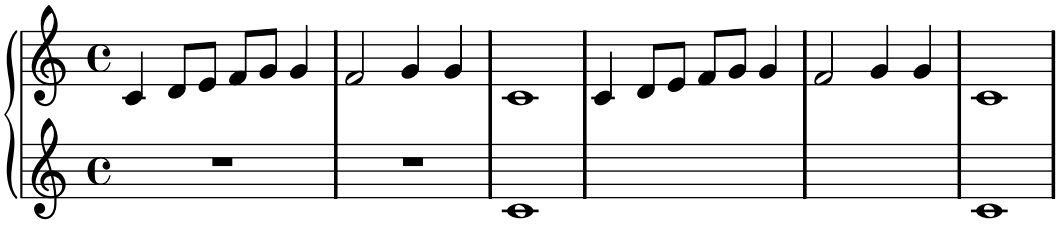
En un contexto orquestal, las cadenzas presentan un problema especial: al construir una partitura que tiene una cadenza, todos los demás instrumentos deben saltar tantas notas como la longitud de la cadenza, pues en caso contrario empezarán demasiado pronto o tarde.

Una solución a este problema son las funciones `mmrest-of-length` y `skip-of-length`. Estas funciones de Scheme toman un fragmento de música como argumento y generan un `\skip` o silencio multicompás, de la longitud exacta del fragmento.

```
miCadencia = \relative c' {
  c4 d8 e f g g4
  f2 g4 g
}

\new GrandStaff <<
  \new Staff {
```

```
    \miCadencia c'1
    \miCadencia c'1
  }
  \new Staff {
    #(ly:export (mmrest-of-length miCadencia))
    c'1
    #(ly:export (skip-of-length miCadencia))
    c'1
  }
>>
```



Véase también

Glosario musical: Sección “cadenza” in *Glosario Musical*.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Gestión del tiempo

El tiempo está administrado por el `Timing_translator`, que de forma predeterminada vive en el contexto de `Score`. Se añade un alias, `Timing`, al contexto en que se coloca el `Timing_translator`.

Se usan las siguientes propiedades de `Timing` para seguir la pista del tiempo dentro de la partitura.

`currentBarNumber`
: El número de compás en curso. Para ver un ejemplo que muestra el uso de esta propiedad, consulte [Numeración de compases], página 73.
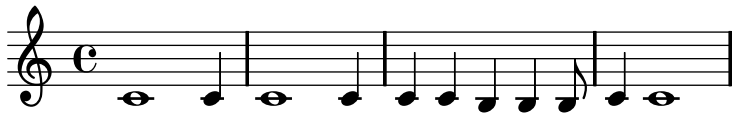
`measureLength`
: La longitud de los compases dentro de la indicación actual de compás. Para un 4/4 esto es 1, y para el 6/8 es 3/4. Su valor determinad cuándo se insertan las líneas divisorias y cómo se generan las barras automáticas.

`measurePosition`
: El punto en que nos encontramos dentro del compás. Esta cantidad se reinicia sustrayendo `measureLength` cada vez que se alcanza o se excede `measureLength`. Cuando eso ocurre, se incrementa `currentBarNumber`.

`timing`
: Si tiene un valor verdadero, las variables anteriores se actualizan a cada paso de tiempo. Cuando tiene un valor falso, el grabador se queda en el compás actual indefinidamente.

La cuenta del tiempo se puede cambiar estableciendo el valor de cualquiera de estas variables explícitamente. En el siguiente ejemplo, se imprime la indicación de compás predeterminada 4/4, pero `measureLength` tiene está ajustado a 5/4. En los 4/8 hasta el tercer compás, la posición `measurePosition` se adelanta en 1/8 hasta 5/8, acortando ese compás en 1/8. Entonces, la siguiente línea divisoria cae en 9/8 en vez de hacerlo en 5/4.

```
\set Score.measureLength = #(ly:make-moment 5 4)
c1 c4
c1 c4
c4 c4
\set Score.measurePosition = #(ly:make-moment 5 8)
b4 b4 b8
c4 c1
```



Como lo ilustra el ejemplo, `ly:make-moment n m` construye una duración de la fracción n/m de una redonda. Por ejemplo, `ly:make-moment 1 8` es una duración de una corchea y `ly:make-moment 7 16` es la duración de siete semicorcheas.
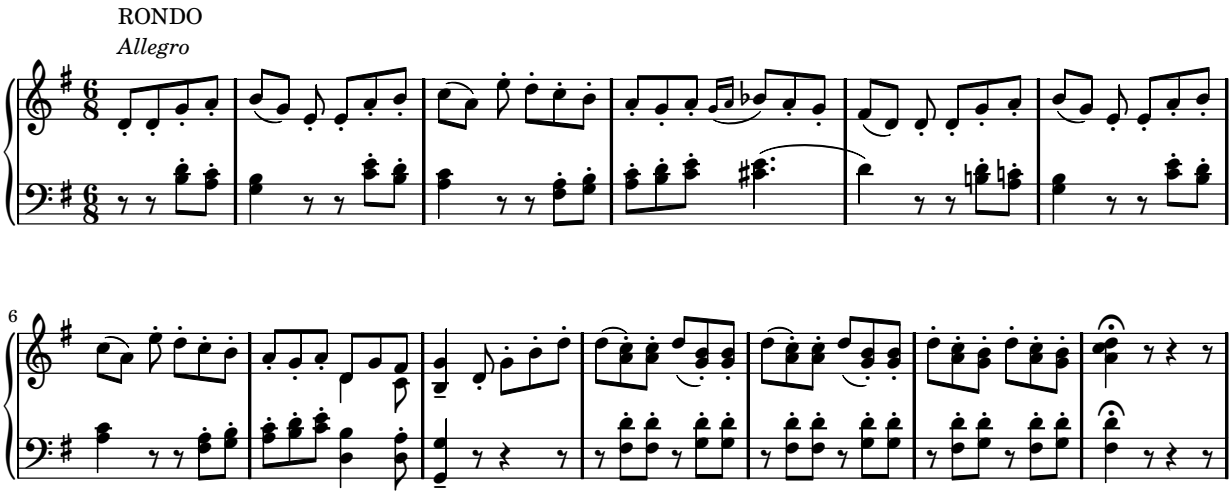
Véase también

Referencia de la notación: [Numeración de compases], página 73, [Música sin compasear], página 49.

Fragmentos de código: Sección “Rhythms” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Timing_translator” in *Referencia de Funcionamiento Interno*, Sección “Score” in *Referencia de Funcionamiento Interno*.

1.3 Expresiones



Esta sección relaciona diversas marcas de expresión que se pueden crear en un a partitura.

1.3.1 Adosado a las notas

Esta sección explica cómo crear marcas expresivas que están aplicadas a notas: articulaciones, ornamentos y matices. También se tratan los métodos para crear las nuevas marcas dinámicas.
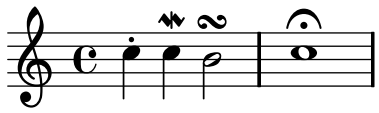
Articulaciones y ornamentos

Se pueden adjuntar a las notas un amplio abanico de símbolos para denotar articulaciones, adornos y otras indicaciones de ejecución, utilizando la sintaxis siguiente:

*nota**nombre*
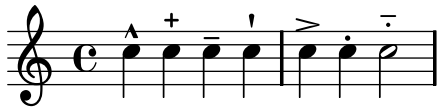
La lista de los posibles valores de *nombre* está en Sección B.10 [Lista de articulaciones], página 513. Por ejemplo:

```
c4\staccato c\mordent b2\turn
c1\fermata
```



Algunas de estas articulaciones tienen abreviaturas que facilitan su escritura. Las abreviaturas se escriben detrás del nombre de la nota, y su sintaxis consiste en un guión `-` seguido de un símbolo que especifica la articulación. Existen abreviaturas predefinidas para el *marcato*, *stopped* (nota apagada), *tenuto*, *staccatissimo*, *accent* (acento), *staccato* (picado), y *portato*. La salida correspondiente a estas articulaciones aparece de la siguiente manera:

```
c4-^  c-+  c--  c-|
c4->  c-.  c2-_
```



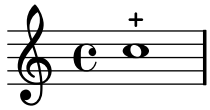
Las reglas para la colocación predeterminada de las articulaciones se encuentran definidas en el archivo '`scm/script.scm`'. Las articulaciones y ornamentos se pueden colocar manualmente por encima o por debajo del pentagrama, véase Sección 5.4.2 [Dirección y posición], página 407.

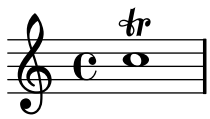
Fragmentos de código seleccionados

Modificar los valores predeterminados para la notación abreviada de las articulaciones

Las abreviaturas se encuentran definidas dentro del archivo `ly/script-init.ly`, donde las variables `dashHat`, `dashPlus`, `dashDash`, `dashBar`, `dashLarger`, `dashDot` y `dashUnderscore` reciben valores predeterminados. Se pueden modificar estos valores predeterminados para las abreviaturas. Por ejemplo, para asociar la abreviatura `-+` (`dashPlus`) con el símbolo del semitrino en lugar del símbolo predeterminado `+`, asigne el valor `trill` a la variable `dashPlus`:

```
\relative c'' { c1-+ }
dashPlus = "trill"
\relative c'' { c1-+ }
```
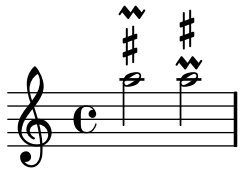




Controlar la ordenación vertical de las inscripciones

El orden vertical que ocupan las inscripciones gráficas está controlado con la propiedad `script-priority`. Cuanto más bajo es este número, más cerca de la nota se colocará. En este ejemplo, el `TextScript` (el sostenido) tiene primero la prioridad más baja, por lo que se sitúa

en la posición más baja en el primer ejemplo. En el segundo, el semitrino (el `Script`) es el que la tiene más baja, por lo que se sitúa en la parte interior. Cuando dos objetos tienen la misma prioridad, el orden en que se introducen determina cuál será el que aparece en primer lugar.

```
\relative c''' {
  \once \override TextScript #'script-priority = #-100
  a2^\prall^\markup { \sharp }

  \once \override Script #'script-priority = #-100
  a2^\prall^\markup { \sharp }
}
```



Véase también

Glosario musical: Sección “tenuto” in *Glosario Musical*, Sección “acento” in *Glosario Musical*, Sección “staccato” in *Glosario Musical*, Sección “portato” in *Glosario Musical*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 407, Sección B.10 [Lista de articulaciones], página 513, [Trinos], página 100.

Archivos instalados: ‘`scm/script.scm`’.

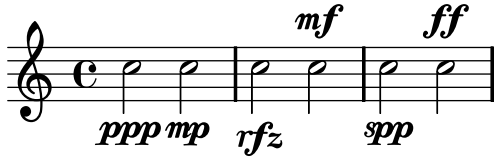
Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Script” in *Referencia de Funcionamiento Interno*, Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Matices dinámicos

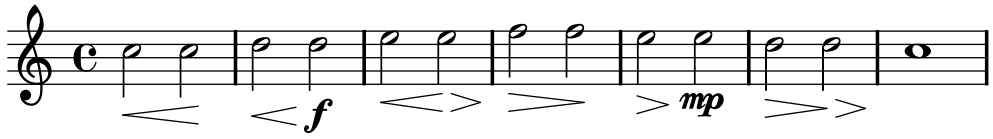
Las marcas dinámicas de matiz absoluto se especifican usando un comando después de una nota: `c4\ff`. Las marcas dinámicas disponibles son `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz` y `\rfz`. Las indicaciones dinámicas se pueden colocar manualmente por encima o por debajo del pentagrama; para ver más detalles, consulte Sección 5.4.2 [Dirección y posición], página 407.

```
c2\ppp c\mp
c2\rfz c^\mf
c2_\spp c^\ff
```



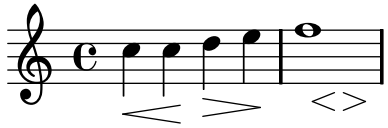
Una indicación de *crescendo* se comienza con `\<` y se termina con `\!`, con un matiz absoluto o con otra indicación de crescendo o de decrescendo. Una indicación de *decrescendo* comienza con `\>` y se termina también con `\!`, con un matiz dinámico absoluto o con otra indicación de crescendo o de decrescendo. Se pueden usar `\cr` y `\decr` en lugar de `\<` y `\>`. De forma predeterminada, se tipografían reguladores en ángulo cuando se utiliza esta notación.

```
c2\< c\!
d2\< d\f
e2\< e\>
f2\> f\!
e2\> e\mp
d2\> d\>
c1\!
```
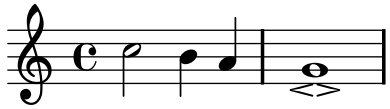


Se requieren silencios espaciadores para tipografiar varias indicaciones dinámicas sobre una sola nota.

```
c4\< c\! d\> e\!
<< f1 { s4 s4\< s4\> s4\! } >>
```
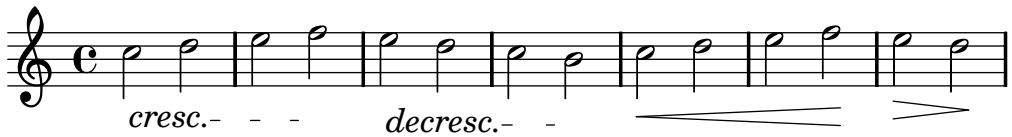


En algunas situaciones, la marca de articulación `\espressivo` puede ser apropiada para indicar un crescendo y decrescendo sobre una nota:

```
c2 b4 a
g1\espressivo
```



Los crescendi y decrescendi se pueden tipografiar como marcas textuales en lugar de reguladores. Las instrucciones incorporadaas que activan estos modos de texto son `\crescTextCresc`, `\dimTextDecresc`, `\dimTextDecr` y `\dimTextDim`. Las instrucciones correspondientes `\crescHairpin` y `\dimHairpin` hacen que vuelvan a aparecer los reguladores de nuevo:

```
\crescTextCresc
c2\< d | e f\!
\dimTextDecresc
e2\> d | c b\!
\crescHairpin
c2\< d | e f\!
\dimHairpin
e2\> d\!
```



Para crear nuevas indicaciones de matiz absoluto o de texto que deba alinearse con los matices, véase [Indicaciones dinámicas contemporáneas], página 90.

La colocación vertical de las indicaciones de dinámica se maneja por parte de Sección "DynamicLineSpanner" in *Referencia de Funcionamiento Interno*.

Instrucciones predefinidas

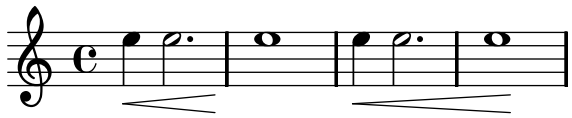
`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`, `\crescTextCresc`, `\dimTextDim`, `\dimTextDecr`, `\dimTextDecresc`, `\crescHairpin`, `\dimHairpin`.

Fragmentos de código seleccionados

Establecer el comportamiento de los reguladores en las barras de compás

Si la nota que da fin a un regulador cae sobre la primera parte de un compás, el regulador se detiene en la línea divisoria inmediatamente precedente. Se puede controlar este comportamiento sobreescribiendo la propiedad `to-barline`.

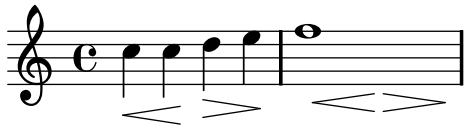
```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin #'to-barline = ##f
  e4\< e2.
  e1\!
}
```



Ajustar la longitud mínima de los reguladores

Si los reguladores son demasiado cortos, se pueden alargar modificando la propiedad `minimum-length` del objeto `Hairpin`.
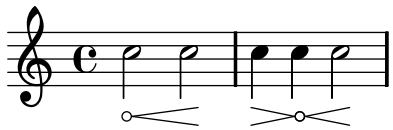
```
\relative c'' {
  c4\< c\! d\> e\!
  \override Hairpin #'minimum-length = #5
  << f1 { s4 s\< s\> s\! } >>
}
```



Impresión de reguladores utilizando la notación «al niente»

Se pueden imprimir reguladores con un círculo en la punta (notación «al niente») estableciendo la propiedad `circled-tip` del objeto `Hairpin` al valor `#t`.

```
\relative c'' {
  \override Hairpin #'circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```



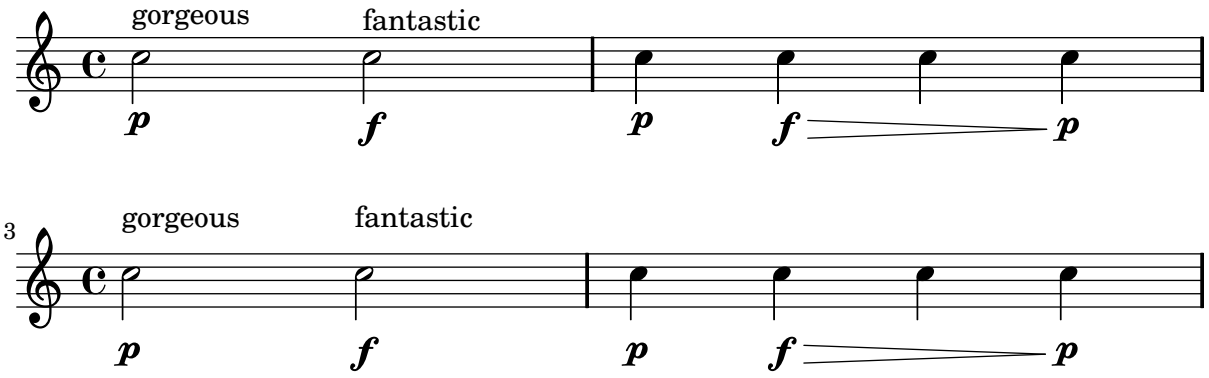
Indicaciones dinámicas y textuales alineadas verticalmente

Mediante el establecimiento de la propiedad `Y-extent` a un valor adecuado, todos los objetos `DynamicLineSpanner` (reguladores e indicaciones dinámicas textuales) se pueden alinear con un punto de referencia común, independientemente de sus dimensiones. De esta manera, todos los elementos estarán alineados verticalmente, produciendo así un resultado más satisfactorio.

Se usa una idea similar para alinear las inscripciones de texto a lo largo de su línea de base.

```
musica = \relative c'' {
  c2\p^\markup { gorgeous } c\f^\markup { fantastic }
  c4\p c\f\> c c\!\p
}

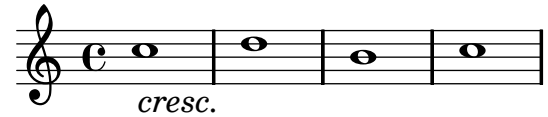
{
  \musica \break
  \override DynamicLineSpanner #'staff-padding = #2.0
  \override DynamicLineSpanner #'Y-extent = #'(-1.5 . 1.5)
  \override TextScript #'Y-extent = #'(-1.5 . 1.5)
  \musica
}
```



Ocultar la línea de extensión de las expresiones textuales de dinámica

Los cambios de dinámica con estilo de texto (como cresc. y dim.) se imprimen con una línea intermitente que muestra su alcance. Esta línea se puede suprimir de la siguiente manera:
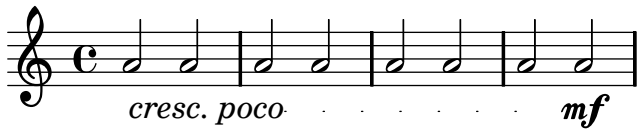
```
\relative c'' {
  \override DynamicTextSpanner #'dash-period = #-1.0
  \crescTextCresc
  c1\< | d | b | c\!
}
```



Cambiar el texto y los estilos de objeto de extensión para las indicaciones dinámicas textuales

Se puede modificar el texto empleado para los crescendos y decrescendos modificando las propiedades de contexto `crescendoText` y `decrescendoText`. El estiloo de la línea de extensión se puede cambiar modificando la propiedad `'style` de `DynamicTextSpanner`. El valor predeterminado es `'hairpin` (regulador), y entre otros valores posibles se encuentran `'line` (línea), `'dashed-line` (línea intermitente) y `'dotted-line` (línea de puntos):

```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner #'style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Véase también

Glosario musical: Sección “al niente” in *Glosario Musical*, Sección “crescendo” in *Glosario Musical*, Sección “decrescendo” in *Glosario Musical*, Sección “regulador” in *Glosario Musical*.

Manual de aprendizaje: Sección “Articulation and dynamics” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 407, [Indicaciones dinámicas contemporáneas], página 90, Sección 3.5.3 [¿Qué sale por el MIDI?], página 338, Sección 3.5.5 [Control de los matices en el MIDI], página 339.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.
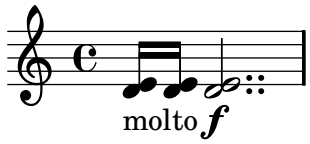
Referencia de funcionamiento interno: Sección “DynamicText” in *Referencia de Funcionamiento Interno*, Sección “Hairpin” in *Referencia de Funcionamiento Interno*, Sección “DynamicLineSpanner” in *Referencia de Funcionamiento Interno*.

Indicaciones dinámicas contemporáneas

La manera más fácil de crear indicaciones dinámicas es usar objetos de marcado (`\markup`).

```
moltoF = \markup { molto \dynamic f }

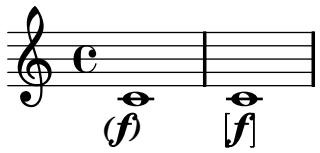
\relative c' {
  <d e>16_\moltoF <d e>
  <d e>2..
}
```



En el modo de marcado se pueden crear indicaciones dinámicas editoriales (entre paréntesis o corchetes). La sintaxis del modo de marcado se describe en Sección 1.8.2 [Formatear el texto], página 173.
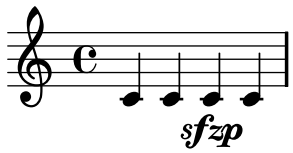
```
f_redondo = \markup { \center-align \concat { \bold { \italic ( }
           \dynamic f \bold { \italic ) } } }
f_rectangulo = \markup { \bracket { \dynamic f } }
\relative c' {
  c1_\f_redondo
  c1_\f_rectangulo
```

```
}
```



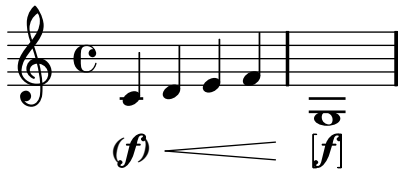
Las indicaciones dinámicas sencillas y centradas se crean fácilmente con la función `make-dynamic-script`. La fuente tipográfica de matices sólo contiene los caracteres `f`, `m`, `p`, `r`, `s` y `z`.

```
sfzp = #(make-dynamic-script "sfzp")
\relative c' {
  c4 c c\sfzp c
}
```
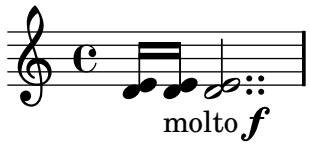


En general `make-dynamic-script` toma cualquier objeto de marcado como argumento. En el siguiente ejemplo, el uso de `make-dynamic-script` asegura la alineación de los objetos de marcado y reguladores que están aplicados sobre la misma nota.

```
f_redondo = \markup { \center-align \concat {
            \normal-text { \bold { \italic ( } }
            \dynamic f
            \normal-text { \bold { \italic ) } } } }
f_rectangulo = \markup { \bracket { \dynamic f } }
dinamicas_f_redondo = #(make-dynamic-script f_redondo)
dinamicas_f_rectangulo = #(make-dynamic-script f_rectangulo)
\relative c' {
  c4_\dinamicas_f_redondo\< d e f
  g,1_\dinamicas_f_rectangulo
}
```



Se puede utilizar en su lugar la forma Scheme del modo de marcado. Su sintaxis se explica en Sección 6.4.1 [Construcción del marcado en Scheme], página 443.

```
moltoF = #(make-dynamic-script
            (markup #:normal-text "molto"
                    #:dynamic "f"))
\relative c' {
  <d e>16 <d e>
  <d e>2..\moltoF
}
```



Los ajustes para las fuentes tipográficas en el modo de marcado se describen en [Instrucciones de marcado de texto más usuales], página 175.

Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 173, [Instrucciones de marcado de texto más usuales], página 175, Sección 6.4.1 [Construcción del marcado en Scheme], página 443, Sección 3.5.3 [¿Qué sale por el MIDI?], página 338, Sección 3.5.5 [Control de los matices en el MIDI], página 339.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.
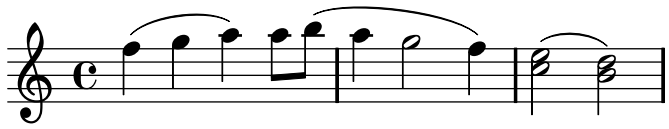
1.3.2 Curvas

Esta sección explica cómo crear varias marcas expresivas de forma curva: ligaduras de expresión y de fraseo, respiraciones, caídas y elevaciones de tono.

Ligaduras de expresión
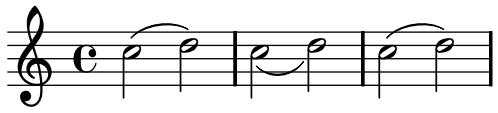
Las *Ligaduras de expresión* se introducen utilizando paréntesis:

```
f4( g a) a8 b(
a4 g2 f4)
<c e>2( <b d>2)
```



Las ligaduras de expresión se pueden colocar manualmente por encima o por debajo de las notas, véase Sección 5.4.2 [Dirección y posición], página 407.

```
c2( d)
\slurDown
c2( d)
\slurNeutral
c2( d)
```
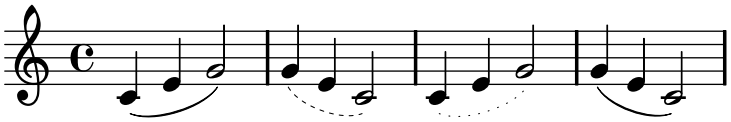


No están permitidas las ligaduras de expresión simultáneas o solapadas, pero una ligadura de fraseo se puede solapar con una ligadura de expresión. Esto hace posible imprimir dos ligaduras distintas al mismo tiempo. Para ver más detalles, consulte [Ligaduras de fraseo], página 93.

Las ligaduras de expresión pueden ser continuas, discontinuas o de puntos. El estilo predeterminado de las ligaduras de expresión es el continuo:

```
c4( e g2)
\slurDashed
g4( e c2)
\slurDotted
c4( e g2)
\slurSolid
```

```
g4( e c2)
```



Instrucciones predefinidas

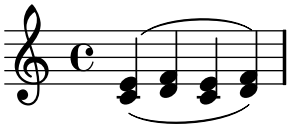
`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurSolid`.

Fragmentos de código seleccionados

Utilizar ligaduras dobles para acordes legato

Algunos compositores escriben dos ligaduras cuando quieren acordes legato. Esto se puede conseguir estableciendo `doubleSlurs`.

```
\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}
```



Véase también

Glosario musical: Sección “ligadura de expresión” in *Glosario Musical*.

Manual de aprendizaje: Sección “On the un-nestedness of brackets and ties” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 407, [Ligaduras de fraseo], página 93.
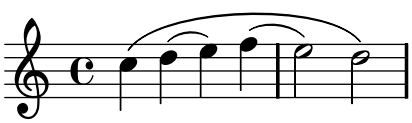
Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Slur” in *Referencia de Funcionamiento Interno*.

Ligaduras de fraseo

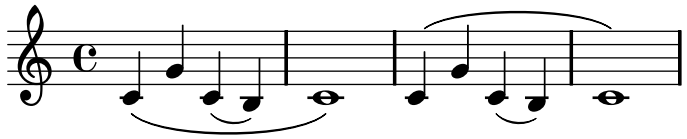
Las *ligaduras de fraseo* (o marcas de fraseo) que indican una frase musical se escriben usando las instrucciones `\(` y `\)` respectivamente:

```
c4\( d( e) f(
e2) d\)
```



Tipográficamente, la ligadura de fraseo se comporta casi exactamente igual que una ligadura de expresión normal. Sin embargo, se tratan como objetos diferentes. Una `\slurUp` no tendrá ningún efecto sobre una ligadura de fraseo. Las ligaduras de fraseo se pueden situar manualmente por encima o por debajo de las notas, véase Sección 5.4.2 [Dirección y posición], página 407.
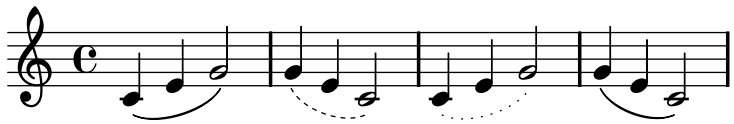
```
c4\( g' c,( b) | c1\)
\phrasingSlurUp
c4\( g' c,( b) | c1\)
```



No se pueden tener varias ligaduras de fraseo simultáneas o solapadas.

Las ligaduras de fraseo pueden ser continuas, de puntos o de rayas. El estilo predeterminado para las ligaduras de fraseo es el continuo:

```
c4\( e g2\)
\phrasingSlurDashed
g4\( e c2\)
\phrasingSlurDotted
c4\( e g2\)
\phrasingSlurSolid
g4\( e c2\)
```



Instrucciones predefinidas

`\phrasingSlurUp`, `\phrasingSlurDown`, `\phrasingSlurNeutral`, `\phrasingSlurDashed`, `\phrasingSlurDotted`, `\phrasingSlurSolid`.

Véase también

Manual de aprendizaje: Sección “On the un-nestedness of brackets and ties” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 407.
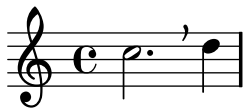
Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “PhrasingSlur” in *Referencia de Funcionamiento Interno*.

Marcas de respiración

Las respiraciones se introducen utilizando `\breathe`:

```
c2. \breathe d4
```



Están contemplados los indicadores musicales para las marcas de respiración en la notación antigua, o divisiones. Para ver más detalles, consulte [Divisiones], página 296.

Fragmentos de código seleccionados

Cambiar el símbolo de la marca de respiración

El glifo de la marca de respiración se puede ajustar sobreescribiendo la propiedad de texto del objeto de presentación `BreathingSign`, con cualquier otro texto de marcado.

```
\relative c'' {
  c2
  \override BreathingSign #'text = \markup { \musicglyph #"scripts.rvarcomma" }
  \breathe
  d2
}
```
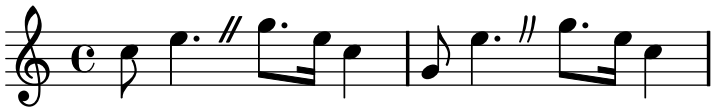


Insertar una cesura

Las marcas de cesura se pueden crear sobreescribiendo la propiedad `'text` del objeto `BreathingSign`. También está disponible una marca de cesura curva.

```
\relative c'' {
  \override BreathingSign #'text = \markup {
    \musicglyph #"scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign #'text = \markup {
    \musicglyph #"scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```



Véase también

Glosario musical: Sección “cesura” in *Glosario Musical*.

Referencia de la notación: [Divisiones], página 296.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BreathingSign” in *Referencia de Funcionamiento Interno*.

Caídas y elevaciones

Se pueden expresar *caídas* y *subidas* de tono (falls y doits) añadidas a las notas mediante la instrucción `\bendAfter`. La dirección de la caída o elevación se indica con un signo más o menos (arriba o abajo). El número indica el intervalo de alturas sobre el que se extiende la caída o elevación *partiendo de* la nota principal.

```
c2-\bendAfter #+4
c2-\bendAfter #-4
c2-\bendAfter #+8
```

```
c2-\bendAfter #-8
```



El guión - que sigue al nombre de la nota es *necesario* al escribir caídas y elevaciones.

Fragmentos de código seleccionados

Ajustar la forma de las subidas y caídas de tono

Puede ser necesario trucar la propiedad `shortest-duration-space` para poder ajustar el tamaño de las caídas y subidas de tono («falls» y «doits»).

```
\relative c'' {
  \override Score.SpacingSpanner #'shortest-duration-space = #4.0
  c2-\bendAfter #+5
  c2-\bendAfter #-3
  c2-\bendAfter #+8
  c2-\bendAfter #-6
}
```



Véase también

Glosario musical: Sección “caída” in *Glosario Musical*, Sección “elevación” in *Glosario Musical*.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.
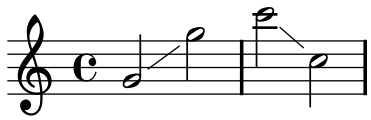
1.3.3 Líneas

Esta sección explica cómo crear diversas marcas de expresión que siguen un camino lineal: glissandi, arpegios y trinos.

Glissando

Se crea un *glissando* adjuntando `\glissando` a la nota:

```
g2\glissando g'
c2\glissando c,
```
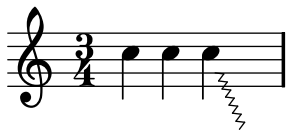


Se pueden crear distintos tipos de glissando. Para ver más detalles, consulte Sección 5.4.7 [Estilos de línea], página 419.

Fragmentos de código seleccionados

Glissando contemporáneo

Se puede tipografiar un gissando contemporáneo sin nota final utilizando una nota oculta y temporalización de cadenza.

```
\relative c'' {
  \time 3/4
  \override Glissando #'style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}
```



Véase también

Glosario musical: Sección “glissando” in *Glosario Musical*.

Referencia de la notación: Sección 5.4.7 [Estilos de línea], página 419.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Glissando” in *Referencia de Funcionamiento Interno*.

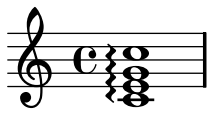
Advertencias y problemas conocidos

La impresión de texto sobre la línea (como *gliss.*) no está contemplada.

Arpegio

Un signo de *acorde arpegiado* (conocido también como acorde quebrado) sobre un acorde, se denota adjuntando `\arpeggio` al acorde:

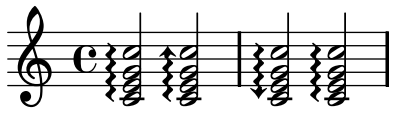
```
<c e g c>1\arpeggio
```



Se pueden escribir distintos tipos de acordes arpegiados. `\arpeggioNormal` produce la vuelta al arpegio normal:

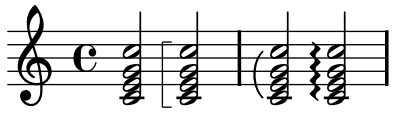
```
<c e g c>2\arpeggio
\arpeggioArrowUp
<c e g c>2\arpeggio
\arpeggioArrowDown
<c e g c>2\arpeggio
```

```
\arpeggioNormal
<c e g c>2\arpeggio
```



Se pueden crear símbolos especiales de arpegio *con corchete*:

```
<c e g c>2
\arpeggioBracket
<c e g c>2\arpeggio
\arpeggioParenthesis
<c e g c>2\arpeggio
\arpeggioNormal
<c e g c>2\arpeggio
```



Los acordes arpegiados se pueden desarrollar explícitamente utilizando *ligaduras de unión*. Véase [Ligaduras de unión], página 36.

Instrucciones predefinidas

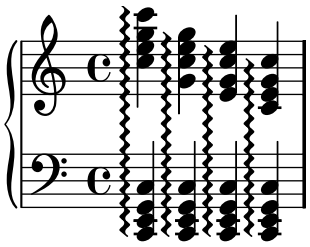
`\arpeggio`, `\arpeggioArrowUp`, `\arpeggioArrowDown`, `\arpeggioNormal`, `\arpeggioBracket`.

Fragmentos de código seleccionados

Crear arpegios que se cruzan entre pentagramas dentro de un sistema de piano

Dentro de un `PianoStaff`, es posible hacer que un arpegio cruce entre los pentagramas ajustando la propiedad `PianoStaff.connectArpeggios`.

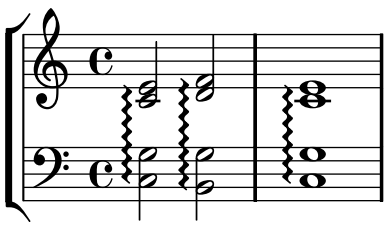
```
\new PianoStaff \relative c'' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
  }
  \new Staff {
    \clef bass
    \repeat unfold 4 {
      <c,, e g c>4\arpeggio
    }
  }
>>
```



Creación de arpegios que se cruzan entre pentagramas dentro de otros contextos

Se pueden crear arpegios que se cruzan entre pentagramas dentro de contextos distintos a `PianoStaff` si se incluye el grabador `Span_arpeggio_engraver` en el contexto de `Score`.

```
\score {
  \new StaffGroup {
    \set Score.connectArpeggios = ##t
    <<
      \new Voice \relative c' {
        <c e>2\arpeggio
        <d f>2\arpeggio
        <c e>1\arpeggio
      }
      \new Voice  \relative c {
        \clef bass
         <c g'>2\arpeggio
         <b g'>2\arpeggio
         <c g'>1\arpeggio
      }
    >>
  }
  \layout {
    \context {
      \Score
      \consists "Span_arpeggio_engraver"
    }
  }
}
```

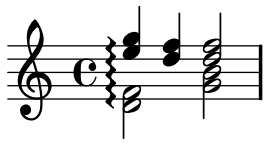


Crear arpegios entre notas de voces distintas

Se puede trazar un símbolo de arpegio entre notas de distintas voces que están sobre el mismo pentagrama si el grabador `Span_arpeggio_engraver` se traslada al contexto de `Staff` context:

```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 } \\
    { <d, f>2\arpeggio <g b>2 }
```

```
  >>
}
```



Véase también

Glosario musical: Sección “arpegio” in *Glosario Musical*.

Referencia de la notación: [Ligaduras de unión], página 36.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Arpeggio” in *Referencia de Funcionamiento Interno*, Sección “PianoStaff” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No es posible mezclar arpegios conectados y no conectados en un `PianoStaff` en el mismo instante de tiempo.

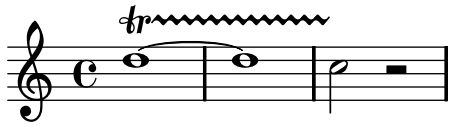
El corchete de los arpegios de paréntesis no funciona para los arpegios de pentagrama cruzado.

Trinos

Los *trinos* cortos sin línea extensora se imprimen con `\trill`; véase [Articulaciones y ornamentos], página 84.
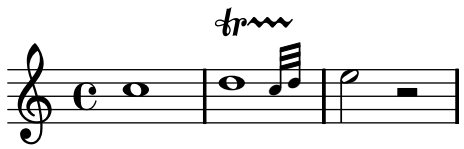
Los *trinos* largos mantenidos, con línea de extensión, se hacen con `\startTrillSpan` y `\stopTrillSpan`:

```
d1~\startTrillSpan
d1
c2\stopTrillSpan r2
```



En el ejemplo siguiente se muestra un trino en combinación con notas de adorno. La sintaxis de esta construcción y el método para efectuar un control más preciso sobre la colocación de las notas de adorno se describe en [Notas de adorno], página 78.

```
c1 \afterGrace
d1\startTrillSpan { c32[ d]\stopTrillSpan }
e2 r2
```
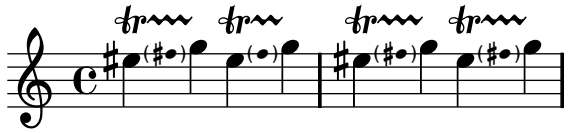


Los trinos que se tienen que ejecutar sobre notas auxiliares explícitas se pueden tipografiar con la instrucción `pitchedTrill`. El primer argumento es la nota principal, y el segndo es la nota *trinada*, que se imprime como una cabeza de nota, sin plica y entre paréntesis.

```
\pitchedTrill e2\startTrillSpan fis
d\stopTrillSpan
```



En el ejemplo siguiente, el segundo trino con nota es ambiguo; la alteración de la nota trinada no se imprime. Como manera de rodear el problema se pueden forzar las alteraciones de las notas trinadas. El segundo compás ejemplifica este método:

```
\pitchedTrill eis4\startTrillSpan fis
g\stopTrillSpan
\pitchedTrill eis4\startTrillSpan fis
g\stopTrillSpan
\pitchedTrill eis4\startTrillSpan fis
g\stopTrillSpan
\pitchedTrill eis4\startTrillSpan fis!
g\stopTrillSpan
```



Instrucciones predefinidas

`\startTrillSpan`, `\stopTrillSpan`.

Véase también
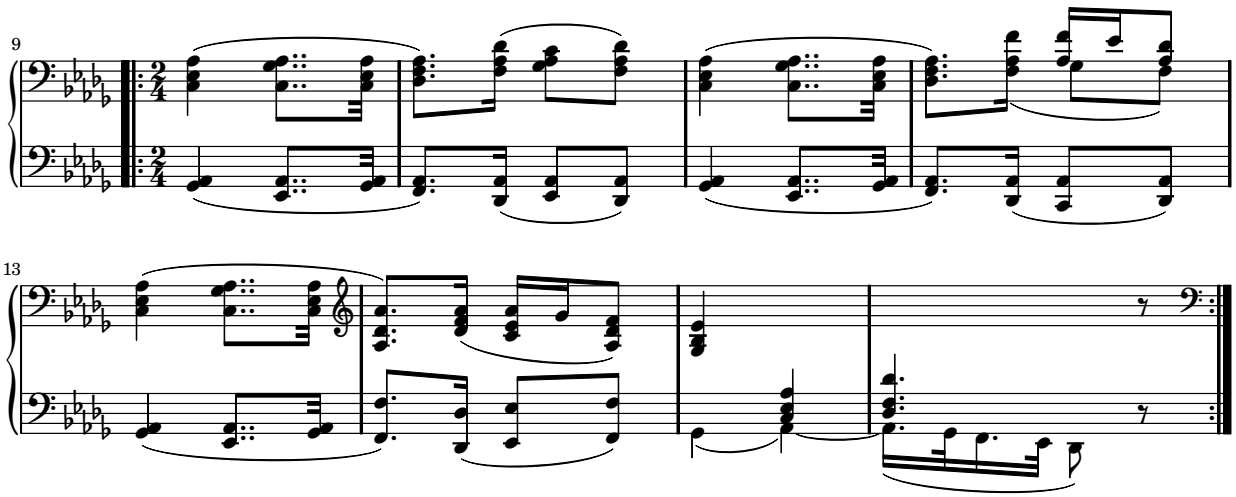
Glosario musical: Sección “trino” in *Glosario Musical*.

Referencia de la notación: [Articulaciones y ornamentos], página 84, [Notas de adorno], página 78.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TrillSpanner” in *Referencia de Funcionamiento Interno*.

1.4 Repeticiones



La repetición es un concepto fundamental en música, y existen varios tipos de notación para las repeticiones. LilyPond contempla los siguientes tipos de repetición:

`volta (primera y segunda vez)`
: La repetición de la música no se imprime de forma desarrollada, sino que se indica encerrándola entre barras de repetición. Si el salto de la repetición se encuentra al comienzo de una pieza, la barra de repetición sólo se imprime al final del fragmento. Se imprimen una serie de finales alternativos (volte) de izquierda a derecha indicados mediante corchetes. Ésta es la notación estándar para las repeticiones con finales alternativos.

`unfold (desplegada)`
: Las música repetida se escribe y se interpreta completamente tantas veces como especifique el valor *número_de_repeticiones*. Es útil cuando se está escribiendo música repetitiva.

`percent (porcentaje)`
: Hacer repeticiones de compases o parte de ellos. Tienen un aspecto semejante a un signo de porcentaje. Las repeticiones de porcentaje se deben declarar dentro de un contexto `Voice`.

`tremolo (trémolo)`
: Hacer barras de trémolo.

1.4.1 Repeticiones largas

Esta sección trata sobre la forma de introducir repeticiones largas, normalmente de varios compases. Las repeticiones adoptan dos formas: repeticiones encerradas entre signos de repetición, o repeticiones explícitas, que se usan para escribir música repetitiva. También se pueden controlar manualmente los signos de repetición.

Repeticiones normales
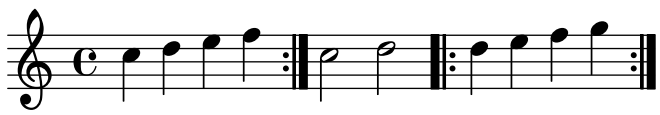
La sintaxis de una repetición simple es

`\repeat` *`variante número_de_repeticiones cuerpo_de_la_repetición`*

donde *cuerpo_de_la_repetición* es una expresión musical. Los finales alternativos se pueden obtener utilizando `\alternative`. Para delimitar los finales alternativos, el grupo de alternativas se debe incluir dentro de un par de llaves. Si existen más repeticiones que finales alternativos, las primeras repeticiones recibirán la primera alternativa.

Repeticiones normales sin finales alternativos:
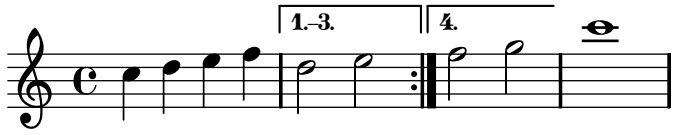
```
\repeat volta 2 { c4 d e f }
c2 d
\repeat volta 2 { d4 e f g }
```



Repeticiones normales con primera y segunda vez:
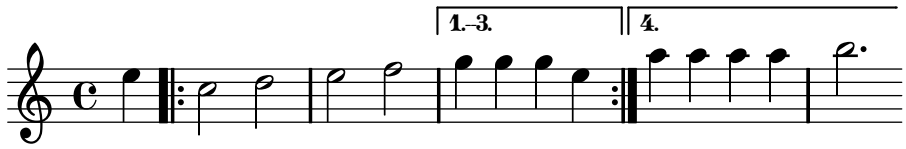
```
\repeat volta 4 { c4 d e f }
\alternative {
  { d2 e }
  { f2 g }
}
```

```
c1
```
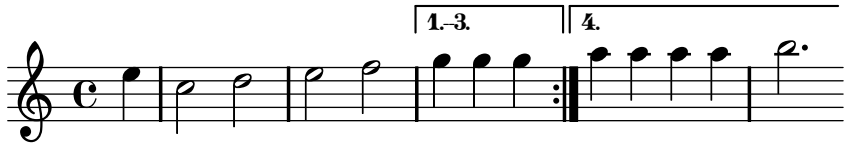


Las repeticiones con anacrusa se pueden introducir de dos formas:

```
\partial 4
e |
\repeat volta 4 { c2 d | e2 f | }
\alternative {
  { g4 g g e }
  { a4 a a a | b2. }
}
```



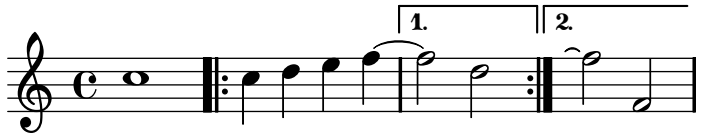
o bien

```
\partial 4
\repeat volta 4 { e4 | c2 d | e2 f | }
\alternative {
  { \partial 4*3 g4 g g }
  { a4 a a a | b2. }
}
```



Se pueden añadir ligaduras de unión al segundo final:

```
c1
\repeat volta 2 { c4 d e f ~ }
\alternative {
  { f2 d }
  { f2\repeatTie f, }
}
```
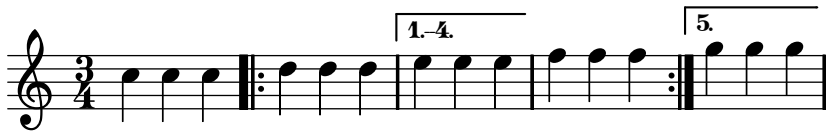


Fragmentos de código seleccionados

Shortening volta brackets

De forma predeterminada, los corchetes de primera y segunda vez se trazan encima de los finales alternativos completos, pero es posible acortartlos estableciendo un valor cierto para `voltaSpannerDuration`. En el ejemplo siguiente, el corchete sólo dura un compás, que corresponde a una duración de 3/4.

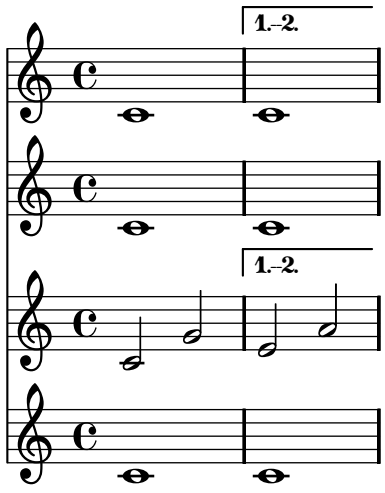
```
\relative c'' {
  \time 3/4
  c4 c c
  \set Score.voltaSpannerDuration = #(ly:make-moment 3 4)
  \repeat volta 5 { d4 d d }
  \alternative {
    {
      e4 e e
      f4 f f
    }
    { g4 g g }
  }
}
```



Añadir corchetes de primera y segunda vez a más pentagramas

El grabador `Volta_engraver` reside de forma predeterminada dentro del contexto de `Score`, y los corchetes de la repetición se imprimen así normalmente sólo encima del pentagrama superior. Esto se puede ajustar añadiendo el grabador `Volta_engraver` al contexto de `Staff` en que deban aparecer los corchetes; véase también el fragmento de código "Volta multi staff".

```
<<
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff \with { \consists "Volta_engraver" } { c'2 g' e' a' }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
>>
```



Véase también

Glosario musical: Sección “repetición” in *Glosario Musical*, Sección “primera y segunda vez” in *Glosario Musical*.

Referencia de la notación: [Barras de compás], página 70, Sección 5.1.3 [Modificar los complementos (plug-ins) de contexto], página 391.

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VoltaBracket” in *Referencia de Funcionamiento Interno*, Sección “RepeatedMusic” in *Referencia de Funcionamiento Interno*, Sección “VoltaRepeatedMusic” in *Referencia de Funcionamiento Interno*, Sección “UnfoldedRepeatedMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Una repetición anidada como

```
\repeat ...
\repeat ...
\alternative
```

es ambigua porque no está claro a qué `\repeat` pertenece la `\alternative`. Esta ambigüedad se resuelve haciendo que la `\alternative` pertenezca siempre a la `\repeat` más interna. Para más claridad, se recomienda usar llaves en tales situaciones.

La información del contador de tiempo no se recuerda al comienzo de una alternativa, por ello después de una repetición la información de la cuenta de tiempo se debe reajustar manualmente; por ejemplo, estableciendo el valor de `Score.measurePosition` o introduciendo `\partial`. De forma similar, las ligaduras de expresión tampoco se repiten.

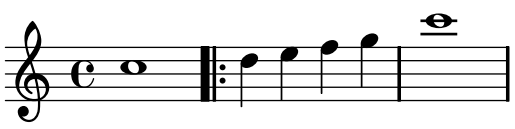
Marcas de repetición manual

Nota: Estos métodos sólo se utilizan para realizar construcciones de repetición poco usuales, y pueden tener un comportamiento distinto al esperado. En casi todas las situaciones, se deben crear las repeticiones utilizando la instrucción estándar `\\repeat` o imprimiendo las barras de compás correspondientes. Para ver más información, consulte [Barras de compás], página 70.

Se puede usar la propiedad `repeatCommands` para controlar la disposición de las repeticiones. Su valor es una lista de Scheme de comandos de repetición.

`start-repeat`
: Imprimir una barra de compás `|:`
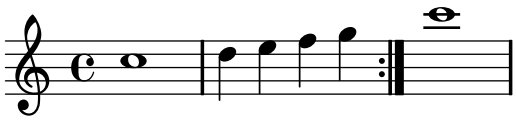
```
c1
\set Score.repeatCommands = #'(start-repeat)
d4 e f g
c1
```



En la práctica habitual del grabado no se imprimen signos de repetición al principio de la pieza.

`end-repeat`
: Imprimir una barra de compás `:|`
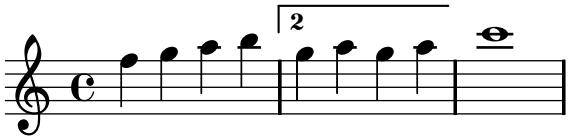
```
c1
d4 e f g
\set Score.repeatCommands = #'(end-repeat)
c1
```



(volta *number*) ... (volta #f)

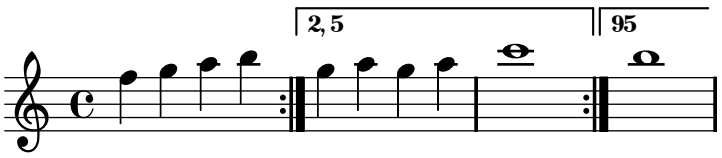
Crear una nueva casilla de repetición con el número que se especifica. El corchete de vez se debe terminar de forma explícita, pues en caso contrario no se imprimirá.

```
f4 g a b
\set Score.repeatCommands = #'((volta "2"))
g4 a g a
\set Score.repeatCommands = #'((volta #f))
c1
```
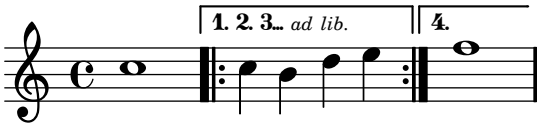


Se pueden producir varias instrucciones de repetición en el mismo punto:

```
f4 g a b
\set Score.repeatCommands = #'((volta "2, 5") end-repeat)
g4 a g a
c1
\set Score.repeatCommands = #'((volta #f) (volta "95") end-repeat)
b1
\set Score.repeatCommands = #'((volta #f))
```



Se puede incluir texto dentro de la casilla de primera y segunda vez. El texto puede ser un número o números, o un elemento de marcado, véase Sección 1.8.2 [Formatear el texto], página 173. La forma más fácil de usar texto de marcado es definir el marcado previamente, y luego incluirlo dentro de una lista de Scheme.

```
voltaAdLib = \markup { 1. 2. 3... \text \italic { ad lib. } }
\relative c'' {
  c1
  \set Score.repeatCommands = #(list(list 'volta voltaAdLib) 'start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```

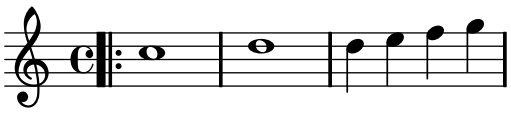


Fragmentos de código seleccionados

Imprimir puntos de repetición al prinicpio de la pieza

Se puede imprimir una línea divisoria de la forma `|:` al principio de la pieza, sobreescribiendo la propiedad correspondiente:

```
\relative c'' {
  \once \override Score.BreakAlignment #'break-align-orders =
        #(make-vector 3 '(instrument-name
                          left-edge
                          ambitus
                          span-bar
                          breathing-sign
                          clef
                          key-signature
                          time-signature
                          staff-bar
                          custos
                          span-bar))
  \bar "|:"
  c1
  d1
  d4 e f g
}
```



Véase también

Referencia de la notación: [Barras de compás], página 70, Sección 1.8.2 [Formatear el texto], página 173.

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VoltaBracket” in *Referencia de Funcionamiento Interno*, Sección “RepeatedMusic” in *Referencia de Funcionamiento Interno*, Sección “VoltaRepeatedMusic” in *Referencia de Funcionamiento Interno*.
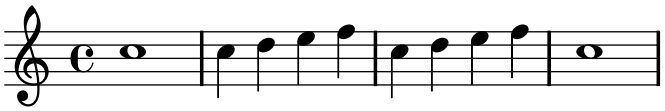
Repeticiones explícitas

Mediante la utilización de la instrucción `unfold` se pueden usar las repeticiones para simplificar la escritura deplegada de música repetitiva. La sintaxis es:

`\repeat unfold` *`número_de_repeticiones expresión_musical`*
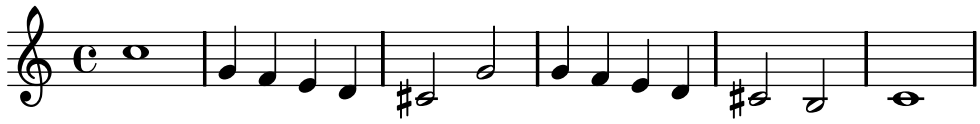
donde *expresión_musical* es una expresión musical y *nnúmero_de_repeticiones* es el número de veces que *expresión_musical* se repite.

```
c1
\repeat unfold 2 { c4 d e f }
c1
```



Se pueden hacer repeticiones desplegadas con finales alternativos. Si hay más repeticiones que finales alternativos, el primer final alternativo se aplica a las primeras repeticiones.

```
c1
\repeat unfold 2 { g4 f e d }
  \alternative {
    { cis2 g' }
    { cis,2 b }
  }
c1
```



Véase también

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RepeatedMusic” in *Referencia de Funcionamiento Interno*, Sección “UnfoldedRepeatedMusic” in *Referencia de Funcionamiento Interno*.

1.4.2 Repeticiones cortas

Esta sección trata de cómo introducir repeticiones cortas. Las repeticiones cortas adoptan dos formas básicas: repeticiones desde una sola nota hasta dos compases representadas por barras inclinadas o símbolos de porcentaje, y trémolos.
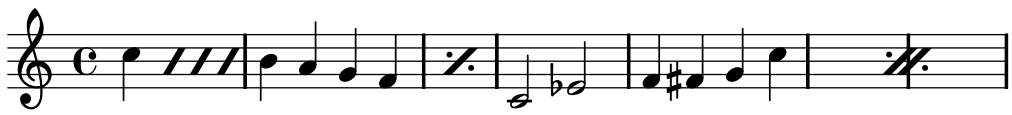
Repeticiones de compás o parte de ellos

Están contempladas las repeticiones de patrones cortos de notas. La música se imprime una sola vez, y el patrón se sustituye por un símbolo especial. Los patrones que son más cortos de un compás se sustituyen por barras inclinadas, y los patrones de uno o dos compases se sustituyen por símbolos de tipo porcentaje. La sintaxis es:

```
\repeat percent número expresión_musical
```

donde *expresión_musical* es una expresión musical.

```
\repeat percent 4 { c4 }
\repeat percent 2 { b4 a g f }
\repeat percent 2 { c2 es | f4 fis g c | }
```

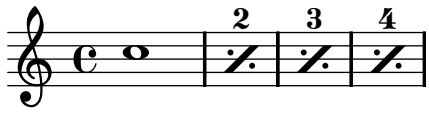


Fragmentos de código seleccionados

Percent repeat counter

Las repeticiones de compases completos mayores de dos repeticiones pueden llevar un contador si se activa la propiedad adecuada, como se ve en este ejemplo:

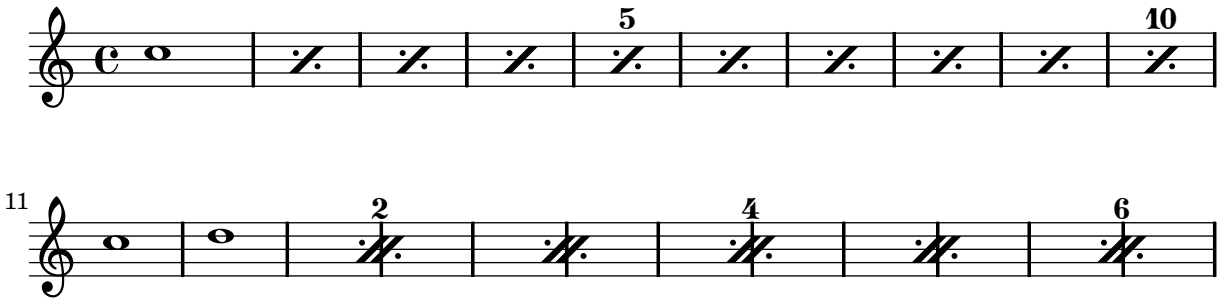
```
\relative c'' {
  \set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```



Visibilidad del contador de repeticiones de tipo porcentaje

Se pueden mostrar los contadores de las repeticiones del tipo porcentaje a intervalos regulares mediante el establecimiento de la propiedad de contexto `repeatCountVisibility`.
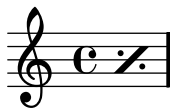
```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```



Símbolos de porcentaje sueltos

También se pueden imprimir símbolos de porcentaje sueltos. Esto se hace introduciendo un silencio multicompás con una función de impresión distinta:

```
\relative c'' {
  \override MultiMeasureRest #'stencil
    = #ly:multi-measure-rest::percent
  \override MultiMeasureRest #'thickness = #0.48
  R1
}
```



Véase también

Glosario musical: Sección “repetición de estilo porcentaje” in *Glosario Musical*, Sección “simile” in *Glosario Musical*.

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RepeatSlash” in *Referencia de Funcionamiento Interno*, Sección “PercentRepeat” in *Referencia de Funcionamiento Interno*, Sección “DoublePercentRepeat” in *Referencia de Funcionamiento Interno*, Sección “DoublePercentRepeatCounter” in *Referencia de Funcionamiento Interno*, Sección “PercentRepeatCounter” in *Referencia de Funcionamiento Interno*, Sección “PercentRepeatedMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos
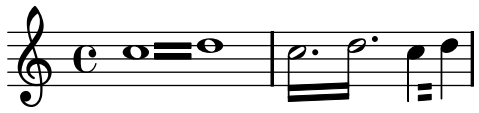
Sólo están contempladas tres clases de repeticiones del tipo porcentaje: una barra inclinada única que representa un solo pulso (sin que importe la duración de las notas repetidas); una sola barra inclinada con puntos que representa un compás completo; y dos barras con puntos que cruzan a una línea divisoria y que representa dos compases completos. No están contempladas ni varias barras inclinadas que representan repeticiones de un pulso consistentes en semicorcheas o figuras menores, ni dos barras con puntos que representan repeticiones de un solo pulso consistentes en notas de duraciones variables.

Repeticiones de trémolo

Los trémolos pueden adoptar dos formas: alternancia entre dos acordes o dos notas, y repetición rápida de una sola nota o acorde. Los trémolos que consisten en una alternancia se indican por medio de la adición de barras entre las notas o acordes que se alternan, mientras que los trémolos que consisten en la repetición rápida de una sola nota se indican mediante la adición de barras cruzadas a una nota única.

Para colocar marcas de trémolo entre las notas, use `\repeat` con el estilo `tremolo` (trémolo):

```
\repeat tremolo 8 { c16 d }
\repeat tremolo 6 { c16 d }
\repeat tremolo 2 { c16 d }
```



La sintaxis de `\repeat tremolo` espera que se escriban exactamente dos notas dentro de las llaves, y el número de repeticiones debe corresponderse con un valor que se pueda expresar con figuras normales o con puntillo. Así, `\repeat tremolo 7` es válido y produce una nota con doble puntillo, pero `\repeat tremolo 9` no es válido.

La duración del trémolo es igual a la duración de la expresión entre llaves, multiplicada por el número de repeticiones: `\repeat tremolo 8 { c16 d16 }` da como resultado un trémolo de redonda, escrito como dos redondas unidas por barras de trémolo.
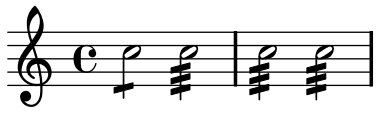
Existen dos maneras de colocar marcas de trémolo sobre una única nota. Incluso aquí se puede utilizar la sintaxis `\repeat tremolo`, en cuyo caso la nota no debe ir encerrada entre llaves:

```
\repeat tremolo 4 c'16
```



El mismo resultado se puede obtener escribiendo ‘`:`[*número*]’ después de la nota. El número indica la duración de la subdivisión, y debe ser 8 como mínimo. Un valor de 8 para *número* produce una línea que atraviesa la plica de la nota. Si se omite la longitud, se utiliza el último valor (almacenado en `tremoloFlags`)

```
c2:8 c:32
c: c:
```
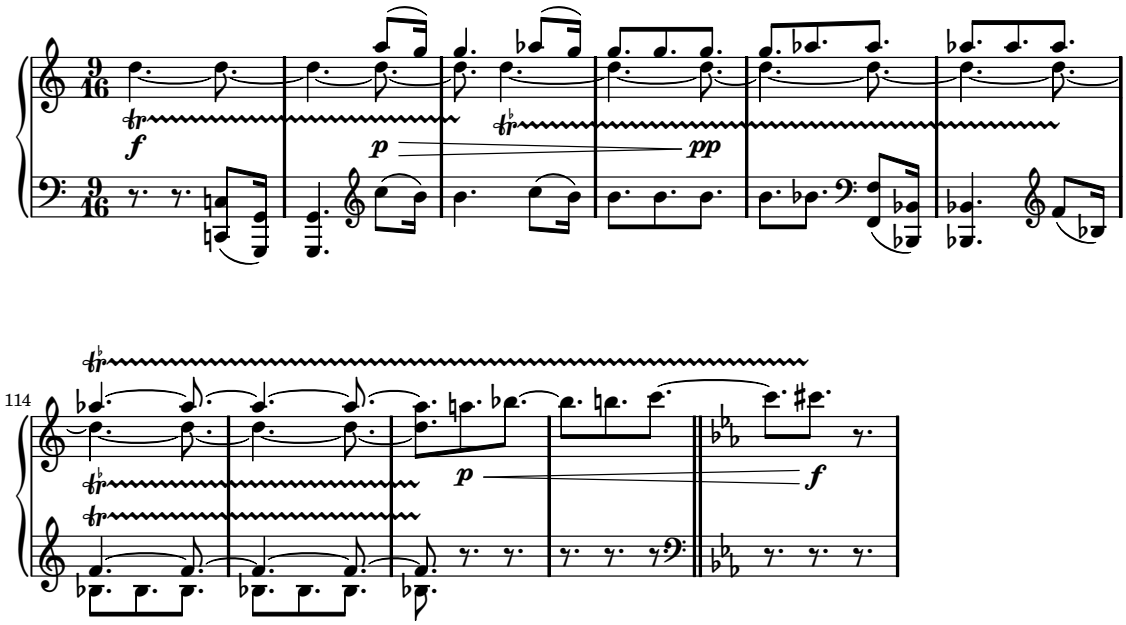


Véase también

Fragmentos de código: Sección “Repeats” in *Fragmentos de código*.

Advertencias y problemas conocidos

Los trémolos de pentagrama cruzado no funcionan bien.

1.5 Notas simultáneas



La polifonía en música hace referencia a tener más de una voz en una pieza cualquiera de música. En LilyPond la polifonía hace referencia a tener más de una voz en el mismo pentagrama.
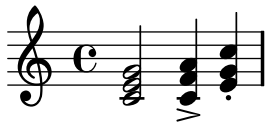
1.5.1 Una voz única

Esta sección trata de la notas simultáneas dentro de la misma voz.

Notas en acorde

Un acorde se forma encerrando un conjunto de notas entre < y >. Un acorde puede ir seguido de una duración o un conjunto de articulaciones, como si fueran simples notas.

```
<c e g>2 <c f a>4-> <e g c>-.
```



Se puede usar el modo relativo para la altura de las notas de los acordes. La octava de cada nota se determina utilizando como referencia la nota precedente, excepto en el caso de la primera nota de un acorde: la referencia para la primera nota es la *primera* nota del acorde anterior.

Para ver más información sobre los acordes, consulte Sección 2.7 [Notación de acordes], página 264.

Véase también

Glosario musical: Sección “acorde” in *Glosario Musical*.

Manual de aprendizaje: Sección “Combining notes into chords” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 2.7 [Notación de acordes], página 264.
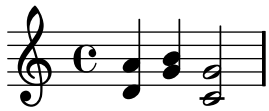
Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Expresiones simultáneas
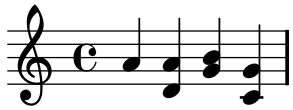
Una o más expresiones musicales encerradas entre ángulos dobles se entienden como simultáneas. Si la primera expresión comienza con una sola nota o si toda la expresión simultánea aparece explícitamente dentro de una sola voz, la expresión completa se sitúa sobre un solo pentagrama; en caso contrario los elementos de la expresión simultánea se sitúan en pentagramas distintos.

Los ejemplos siguientes muestran expresiones simultáneas sobre un solo pentagrama:

```
\new Voice {  % una voz explícita
  << {a4 b g2} {d4 g c,2} >>
}
```



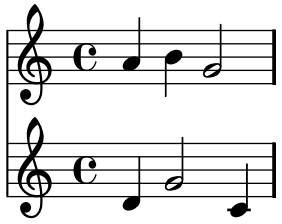
```
% primera nota única
a << {a4 b g}  {d4 g c,} >>
```



Esto puede ser de utilidad si las secciones siimultáneas tienen idénticas duraciones, pero se producirán errores si se intentan poner notas de distinta duración sobre la misma plica.

El ejemplo siguiente muestra cómo las expresiones simultáneas pueden generar varios pentagramas de forma implícita:

```
% primera nota no única
<< {a4 b g2}  {d4 g2 c,4} >>
```

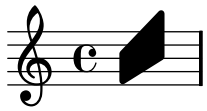


Aquí no hay problema en tener distintas duraciones.

Racimos (clusters)

Un «cluster» o racimo indica que se deben tocar simultáneamente un conjunto de notas consecutivas. Se escriben aplicando la función `\makeClusters` a una secuencia de acordes, p.ej.:

```
\makeClusters { <g b>2 <c g'> }
```



Se pueden mezclar en el mismo pentagrama notas normales y clusters, incluso al mismo tiempo. En tal caso, no se hace ningún intento de evitar automáticamente las colisiones entre clusters y notas normales.

Véase también

Glosario musical: Sección “racimo (cluster)” in *Glosario Musical*.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ClusterSpanner” in *Referencia de Funcionamiento Interno*, Sección “ClusterSpannerBeacon” in *Referencia de Funcionamiento Interno*, Sección “Cluster_spanner_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los clusters sólo tienen un buen aspecto cuando abarcan un mínimo de dos acordes. En caso contrario aparecerán excesivamente estrechos.

Los clusters no llevan plica y por sí mismos no pueden indicar las duraciones, pero la longitud del cluster que se imprime viene determinada por la duración de los acordes que lo definen. Los racimos separados necesitan silencios de separación entre ellos.

Los clusters no producen ninguna salida MIDI.

1.5.2 Varias voces
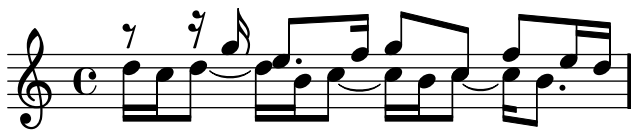
Esta sección trata las notas simultáneas en varias voces o varios pentagramas.

Polifonía en un solo pentagrama

Instanciar las voces explícitamente

La estructura básica necesaria para obtener varias voces independientes en un mismo pentagrama se ilustra en el ejemplo siguiente:

```
\new Staff <<
  \new Voice = "first"
    { \voiceOne r8 r16 g e8. f16 g8[ c,] f e16 d }
  \new Voice= "second"
    { \voiceTwo d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```



Aquí se crean explícitamente instancias de voces, cada una de las cuales recibe un nombre. Las instrucciones `\voiceOne` ... `\voiceFour` (voz uno hasta voz cuatro) preparan las voces de manera que la primera y segunda voces llevan las plicas hacia arriba, las voces segunda y cuarta llevan las plicas hacia abajo, las cabezas de las notas en las voces tercera y cuarta se desplazan horizontalmente, y los silencios de las voces respectivas se desplazan también automáticamente para evitar las colisiones. La instrucción `\oneVoice` (una voz) devuelve todos los ajustes de las voces al estado neutro predeterminado.
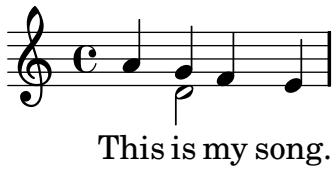
Pasajes polifónicos temporales

Se puede crear un pasaje polifónico temporal con la construcción siguiente:

```
<< { \voiceOne ... }
  \new Voice { \voiceTwo ... }
>> \oneVoice
```

Aquí, la primera expresión dentro de un pasaje polifónico temporal se coloca en el contexto `Voice` que estaba en uso inmediatamente antes del pasaje polifónico, y ese mismo contexto `Voice` continua después de la sección temporal. Otras expresiones dentro de los ángulos se asignan a distintas voces temporales. Esto permite asignar la letra de forma continua a una voz antes, durante y después de la sección polifónica:

```
<<
  \new Voice = "melodia" {
    a4
    <<
      {
        \voiceOne
        g f
      }
      \new Voice {
        \voiceTwo
        d2
      }
    >>
    \oneVoice
    e4
  }
  \new Lyrics \lyricsto "melodia" {
  This is my song.
  }
>>
```
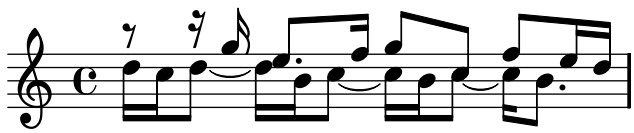


Aquí, las instrucciones `\voiceOne` y `\voiceTwo` son necesarias para definir los ajustes de cada voz.

La construcción de la doble barra invertida

La construcción `<< {...} \\ {...} >>`, en que las dos expresiones (o más) están separadas por doble barra invertida, se comporta de forma distinta a la construcción similar sin las dobles barras: *todas* las expresiones dentro de esta construcción se asignan a contextos `Voice` nuevos. Estos contextos `Voice` nuevos se crean implícitamente y reciben los nombres fijos `"1"`, `"2"`, etc.

El primer ejemplo podría haberse tipografiado de la manera siguiente:
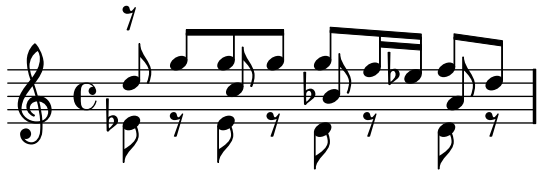
```
<<
  { r8 r16 g e8. f16 g8[ c,] f e16 d }
  \\
  { d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```



Esta sintaxis se puede usar siempre que no nos importe que se creen nuevas voces temporales que después serán descartadas. Estas voces creadas implícitamente reciben ajustes equivalentes al efecto de las instrucciones `\voiceOne` ... `\voiceFour`, en el orden en que aparecen en el código.

En el siguiente ejemplo, la voz intermedia lleva las plicas hacia arriba, de manera que la introducimos en tercer lugar para que pase a ser la voz tres, que tiene las plicas hacia arriba tal y como queremos. Se usan silencios de separación para evitar la aparición de silencios duplicados.

```
<<
  { r8 g g  g g f16 ees f8 d }
  \\
  { ees,8 r ees r d r d r }
  \\
  { d'8 s c s bes s a s }
>>
```
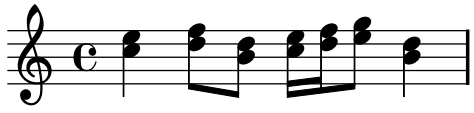


En todas las partituras excepto las más simples, se recomienda crear contextos `Voice` explícitos como aparece explicado en Sección “Contexts and engravers” in *Manual de Aprendizaje* y en Sección “Explicitly instantiating voices” in *Manual de Aprendizaje*.

Duraciones idénticas

En el caso especial en que queremos tipografiar fragmentos de música que discurre en paralelo y con las mismas duraciones, se pueden combinar en un solo contexto de voz, formando así acordes. Para conseguirlo, las incorporamos dentro de una construcción de música simultánea, dentro de una voz creada explícitamente:

```
\new Voice <<
  { e4 f8 d e16 f g8 d4 }
  { c4 d8 b c16 d e8 b4 }
>>
```



Este método conduce a barrados extraños y advertencias si los fragmentos de música no tienen las mismas duraciones exactas.

Instrucciones predefinidas

`\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`, `\oneVoice`.

Véase también

Manual de aprendizaje: Sección “Voices contain music” in *Manual de Aprendizaje*, Sección “Explicitly instantiating voices” in *Manual de Aprendizaje*.

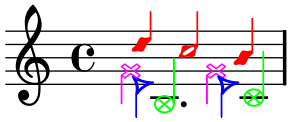
Referencia de la notación: [Pautas de percusión], página 253, [Silencios invisibles], página 41.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Estilos de voz

Se pueden aplicar colores y formas distintos a las voces para permitir identificarlas fácilmente:

```
<<
  { \voiceOneStyle d4 c2 b4 }
  \\
  { \voiceTwoStyle e,2 e }
  \\
  { \voiceThreeStyle b2. c4 }
  \\
  { \voiceFourStyle g'2 g }
>>
```



Para recuperar la presentación normal se utiliza la instrucción `\voiceNeutralstyle`.

Instrucciones predefinidas

`\voiceOneStyle`, `\voiceTwoStyle`, `\voiceThreeStyle`, `\voiceFourStyle`, `\voiceNeutralStyle`.

Véase también
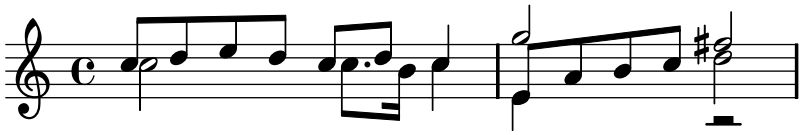
Manual de aprendizaje: Sección “I’m hearing Voices” in *Manual de Aprendizaje*, Sección “Other sources of information” in *Manual de Aprendizaje*.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Resolución de las colisiones
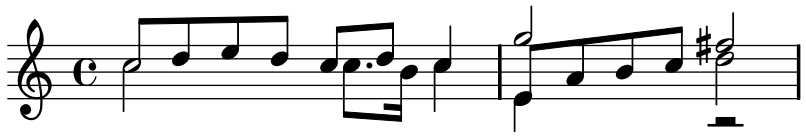
Las cabezas de notas que están en diferentes voces y tienen la misma altura, la misma forma de cabezaa, y dirección opuesta de la plica, se combinan automáticamente, pero las que tienen cabezas distintas o la misma dirección de la plica no se combinan. Los silencios que se encuentran en el lado opuesto de una plica en otra voz se desplazan verticalmente.

```
<<
  {
    c8 d e d c d c4
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```
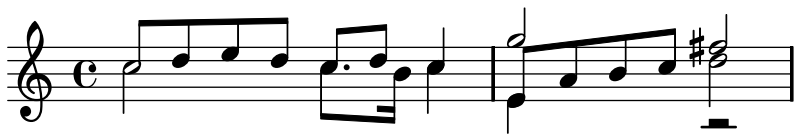


Las cabezas de notas diferentes se pueden combinar, con la excepción de blancas con negras:

```
<<
  {
    \mergeDifferentlyHeadedOn
    c8 d e d c d c4
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



Se pueden combinar cabezas con puntillos diferentes:

```
<<
  {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c8 d e d c d c4
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```
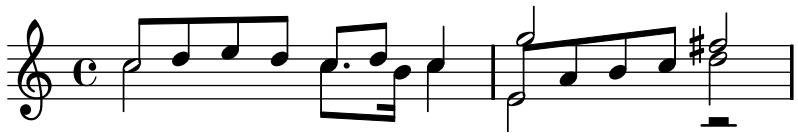


La blanca y la corchea en el comienzo del segundo compás no están correctamente combinadas porque `\mergeDifferentlyHeadedOn` no es capaz de completar satisfactoriamente la combinación cuando se encuentran tres o más notas alineadas en la misma columna, y en este caso se obtiene una advertencia. Para conseguir que la combinación funcione adecuadamente, se debe aplicar un `\shift` o desplazamiento a la nota que no se debe combinar. Aquí, se aplica `\shiftOn` para mover el sol agudo fuera de la columna, y entonces `\mergeDifferentlyHeadedOn` hace su trabajo correctamente.

```
<<
  {
    \mergeDifferentlyHeadedOn
```

```
    \mergeDifferentlyDottedOn
    c8 d e d c d c4
    \shiftOn
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }

>>
```



Las instrucciones `\shiftOn`, `\shiftOnn` y `\shiftOnnn` especifican el grado en que se deben desplazar los acordes de la voz en curso. Las voces exteriores (normalmente: la voz uno y la voz dos) tienen `\shiftOff` (desplazamiento desactivado), mientras que las voces interiores (tres y cuatro) tienen `\shiftOn` (desplazamiento activado). `\shiftOnn` y `\shiftOnnn` definen niveles de desplazamiento más grandes.

Sólo se combinan las notas si tienen la plica en direcciones opuestas (por ejemplo, en las `Voice`s 1 y 2).

Instrucciones predefinidas

`\mergeDifferentlyDottedOn`, `\mergeDifferentlyDottedOff`, `\mergeDifferentlyHeadedOn`, `\mergeDifferentlyHeadedOff`.

`\shiftOn`, `\shiftOnn`, `\shiftOnnn`, `\shiftOff`.
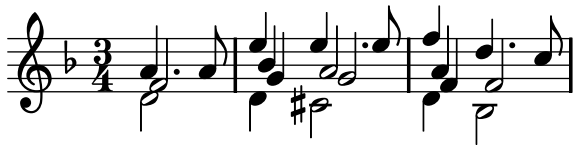
Fragmentos de código seleccionados

Voces adicionales para evitar colisiones

En ciertos casos de polifonía compleja, se necesitan voces adicionales para evitar colisiones entre las notas. Las voces adicionales se añaden definiendo una variable que utiliza la función de Scheme context-spec-music.

```
vozCinco = #(context-spec-music (make-voice-props-set 4) 'Voice)
\relative c'' {
  \time 3/4 \key d \minor \partial 2
  <<
    { \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    } \\ {
      \voiceThree
      f,2
      bes4 a2
      a4 s2
```

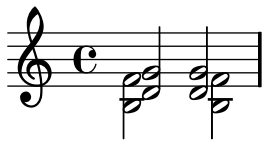
```
    } \\ {
      \vozCinco
      s2
      g4 g2
      f4 f2
    } \\ {
      \voiceTwo
      d2
      d4 cis2
      d4 bes2
    }
  >>
}
```



Forzar el desplazamiento horizontal de las notas

Cuando el motor de tipografiado no es capaz de todo, se puede usar la propiedad force-hshift del objeto NoteColumn para sobreescribir decisiones de tipografiado. Las unidades de medida que se usan aquí son espacios de pentagrama.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
    <b f'>2
    \once \override NoteColumn #'force-hshift = #1.7
    <b f'>2
  }
>>
```



Véase también

Glosario musical: Sección “polifonía” in *Glosario Musical*.

Manual de aprendizaje: Sección “Multiple notes at once” in *Manual de Aprendizaje*, Sección “Voices contain music” in *Manual de Aprendizaje*, Sección “Collisions of objects” in *Manual de Aprendizaje*.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “NoteColumn” in *Referencia de Funcionamiento Interno*, Sección “NoteCollision” in *Referencia de Funcionamiento Interno*, Sección “RestCollision” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Cuando se emplea `\mergeDifferentlyHeadedOn` con una corchea (o una nota más corta) con la plica hacia arriba, y una blanca con la plica hacia abajo, la corchea recibe un desplazamiento ligeramente incorrecto a causa de la anchura diferente del símbolo de la cabeza de la blanca.

No están contemplados los acordes en que la misma nota se presenta con diferentes alteraciones accidentales dentro del mismo. En este caso se recomienda usar la transcripción enarmónica, o usar la notación especial de racimos (véase [Racimos (clusters)], página 112).

Combinación automática de las partes

La combinación automática de particellas se usa para mezclar dos partes musicales sobre un pentagrama. Tiene como objetivo el tipografiado de partituras orquestales. Cuando las dos partes son idénticas durante un período de tiempo, sólo se presenta una de ellas. En los lugares en que las dos partes son diferentes, se tipografían como voces separadas, y las direcciones de las plicas se establecen de forma automática. También las partes de solo y *a due* quedan identificadas y es posible marcarlas.

La sintaxis para la combinación de las partes es:
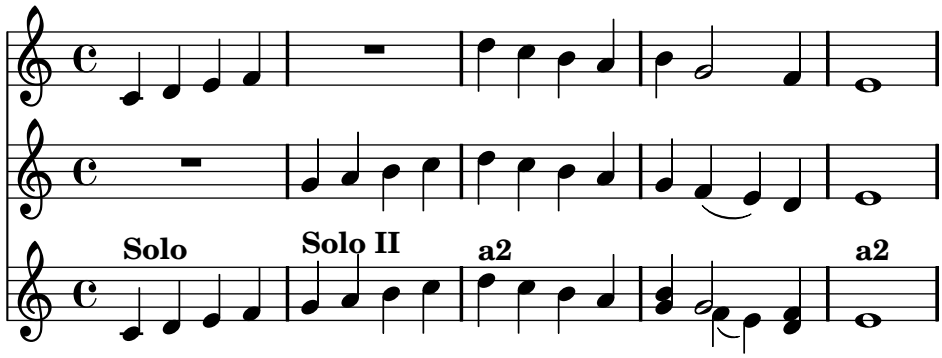
```
\partcombine expresión_musical_1 expresión_musical_2
```

El ejemplo siguiente ejemplifica la funcionalidad básica del combinador de partes: poner las partes en un solo pentagrama, y establecer las direcciones de las plicas y la polifonía. Se utilizan las mismas variables para las partes independientes y el pentagrama combinado.

```
instrumentoUno = \relative c' {
  c4 d e f
  R1
  d'4 c b a
  b4 g2 f4
  e1
}

instrumentoDos = \relative g' {
  R1
  g4 a b c
  d c b a
  g f( e) d
  e1
}

<<
  \new Staff \instrumentoUno
  \new Staff \instrumentoDos
  \new Staff \partcombine \instrumentoUno \instrumentoDos
>>
```



Las notas del tercer compás aparecen solamente una vez a pesar de que se han especificado en las dos partes. Las direcciones de las plicas y ligaduras se establecen de forma automática, según se trate de un solo o de un unísono. Cuando se necesita en situaciones de polifonía, la primera parte (que recibe el nombre de contexto `one`) siempre recibe las plicas hacia arriba, mientras que la segunda (llamada `two`) siempre recibe las plicas hacia abajo. En los fragmentos de solo, las partes se marcan con 'Solo' y 'Solo II', respectivamente. Las partes en unísono (*a due*) se marcan con el texto predeterminado "a2".

Los dos argumentos de `\partcombine` se interpretan como contextos de Sección "Voice" in *Referencia de Funcionamiento Interno*. Si se están usando octavas relativas, se debe especificar `\relative` para ambas expresiones musicales, es decir:

```
\partcombine
  \relative ... expresión_musical_1
  \relative ... expresión_musical_2
```

Una sección `\relative` que se encuentra fuera de `\partcombine` no tiene ningún efecto sobre las notas de *expresión_musical_1* y *expresión_musical_2*.

Fragmentos de código seleccionados

Combinar dos partes sobre el mismo pentagrama

La herramienta de combinación de partes (instrucción `\partcombine`) permite la combinación de varias partes diferentes sobre el mismo pentagrama. Las indicaciones textuales tales como "solo" o "a2" se añaden de forma predeterminada; para quitarlas, sencillamente establezca la propiedad `printPartCombineTexts` al valor "falso". Para partituras vocales (como himnos), no hay necesidad de añadir los textos "solo" o "a2", por lo que se deben desactivar. Sin embargo, podría ser mejor no usarlo si hay solos, porque éstos no se indicarán. En tales casos podría ser preferible la notación polifónica estándar.
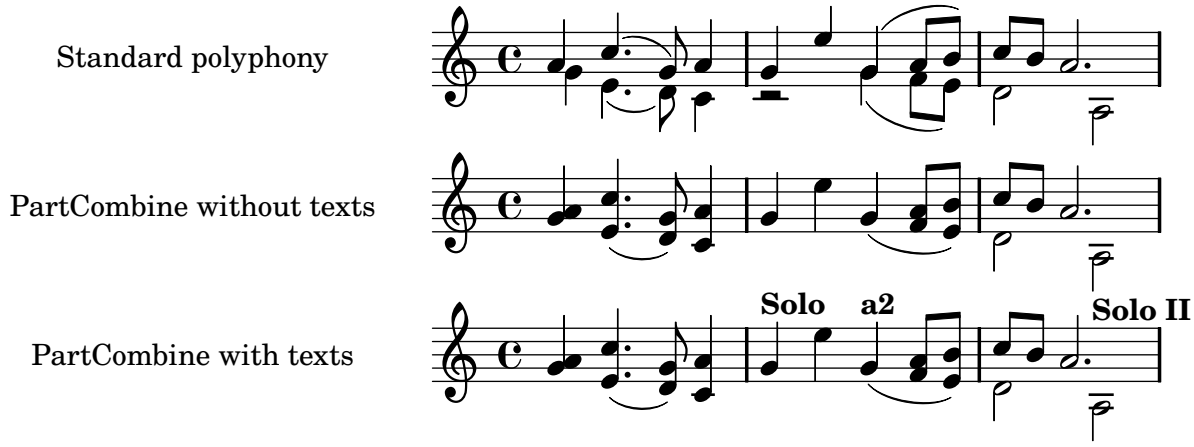
Este fragmento de código presenta las tres formas en que se pueden imprimir dos partes sobre un solo pentagrama: polifonía estándar, `\partcombine` sin textos, y `\partcombine` con textos.

```
musicaArriba = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicaAbajo = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
  <<
```

```
    <<
    \new Staff {
      \set Staff.instrumentName = "Standard polyphony  "
      << \musicaArriba \\ \musicaAbajo >>
    }
    \new Staff \with { printPartCombineTexts = ##f } {
      \set Staff.instrumentName = "PartCombine without texts  "
      \partcombine \musicaArriba \musicaAbajo
    }
    \new Staff {
      \set Staff.instrumentName = "PartCombine with texts  "
      \partcombine \musicaArriba \musicaAbajo
    }
    >>
  >>
  \layout {
    indent = 6.0\cm
    \context {
      \Score
      \override SystemStartBar #'collapse-height = #30
    }
  }
}
```
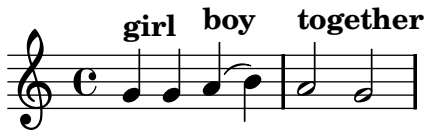


Cambiar los textos de partcombine

Al utilizar la posibilidad de combinación automática de partes, se puede modificar el texto que se imprime para las secciones de solo y de unísono:

```
\new Staff <<
  \set Staff.soloText = #"girl"
  \set Staff.soloIIText = #"boy"
  \set Staff.aDueText = #"together"
  \partcombine
    \relative c'' {
      g4 g r r
      a2 g
    }
    \relative c'' {
      r4 r a( b)
```

```
      a2 g
    }
>>
```



Véase también

Glosario musical: Sección “a due” in *Glosario Musical*, Sección “parte” in *Glosario Musical*.

Referencia de la notación: Sección 1.6.3 [Escritura de las particellas], página 142.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “PartCombineMusic” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

`\partcombine` sólo admite dos voces.

Si `printPartCombineTexts` está establecido, cuando las dos voces tocan y terminan las mismas notas, el combinador de partes puede tipografiar `a2` más de una vez en el mismo compás.

`\partcombine` no puede estar dentro de `\times`.

`\partcombine` no puede estar dentro de `\relative`.

Internamente, el `\partcombine` interpreta los dos argumentos como `Voice`s y decide cuándo se pueden combinar las dos partes. Cuando tienen distintas duraciones, no se combinan y reciben los nombres `one` y `two`. Como consecuencia, si los argumentos cambian a contextos de Sección “Voice” in *Referencia de Funcionamiento Interno* con nombres diferentes, los eventos que contienen se ignorarán. De forma similar, la combinación de partes no está pensada para que funcione con la letra de las canciones; cuando una de las voces recibe un nombre explícito con el objeto de adjuntarle una letra, la combinación de partes deja de funcionar.

`\partcombine` observa solamente los tiempos de ataque de las notas. No puede determinar si una nota iniciada previamente está sonando o no, lo que origina diversos problemas.

Escribir música en paralelo

La música para varias partes se puede intercalar dentro del código de entrada. La función `\parallelMusic` admite una lista que contiene los nombres de las variables que se van a crear, y una expresión musical. El contenido de los compases alternativos extraídos de la expresión se convierten en el valor de las variables respectivas, de manera que podemos utilizarlas más tarde para imprimir la música.
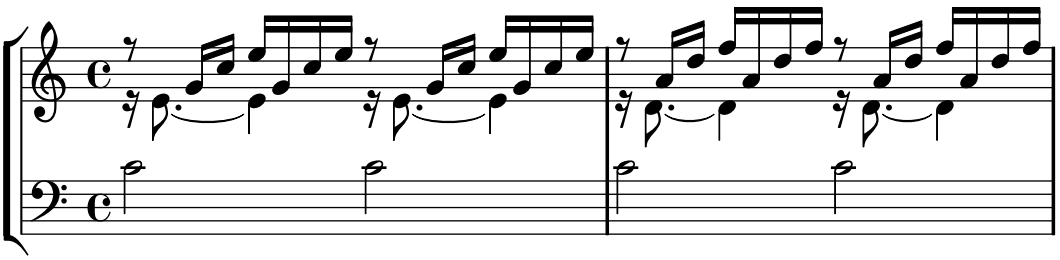
> **Nota:** Es obligatorio utilizar comprobaciones de compás `|`, y los compases deben tener la misma longitud.

```
\parallelMusic #'(voiceA voiceB voiceC) {
  % Compás 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~   e'4           r16 e'8.~   e'4           |
  c'2                       c'2                       |

  % Compás 2
```

```
  r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
  r16 d'8.~   d'4           r16 d'8.~   d'4           |
  c'2                       c'2                       |

}
\new StaffGroup <<
  \new Staff << \voiceA \\ \voiceB >>
  \new Staff { \clef bass \voiceC }
>>
```

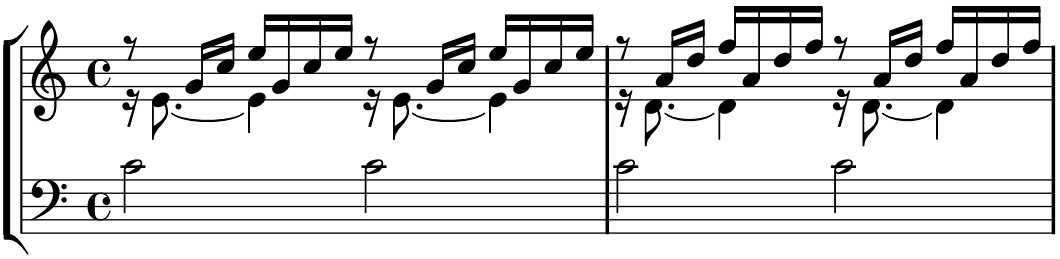


Se puede usar el modo relativo. Observe que la instrucción `\relative` no se utiliza dentro del propio bloque `\parallelMusic`. Las notas guardan relación con la nota anterior en la misma voz, no con la nota anterior dentro del código de entrada (dicho de otra manera, las notas relativas de `vozA` ignoran a las notas que hay en `vozB`.

```
\parallelMusic #'(voiceA voiceB voiceC) {
  % Compás 1
  r8 g16 c e g, c e r8 g,16 c e g, c e  |
  r16 e8.~ e4       r16 e8.~  e4        |
  c2                c                   |

  % Compás 2
  r8 a,16 d f a, d f r8 a,16 d f a, d f |
  r16 d8.~  d4       r16 d8.~  d4       |
  c2                 c                  |

 }
\new StaffGroup <<
  \new Staff << \relative c'' \voiceA \\ \relative c' \voiceB >>
  \new Staff \relative c' { \clef bass \voiceC }
>>
```



Esto funciona aceptablemente bien para música de piano. El siguiente ejemplo asigna cada cuatro compases consecutivos a cuatro variables:

```
global = {
  \key g \major
```

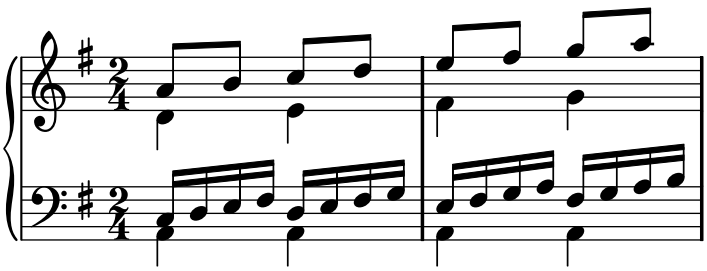
```
  \time 2/4
}

\parallelMusic #'(voiceA voiceB voiceC voiceD) {
  % Compás 1
  a8    b     c   d     |
  d4          e         |
  c16 d e fis d e fis g |
  a4          a         |

  % Compás 2
  e8      fis  g     a   |
  fis4         g         |
  e16 fis g  a fis g a b |
  a4           a         |

  % Compás 3 ...
}
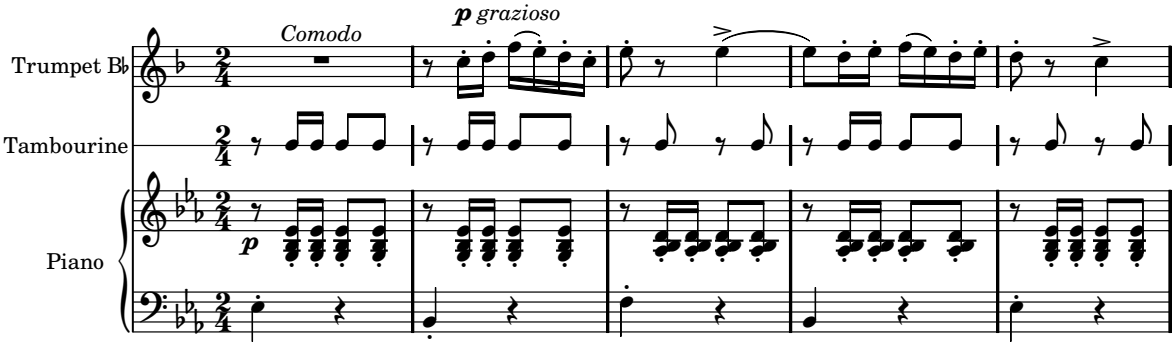
\score {
  \new PianoStaff <<
     \new Staff {
       \global
       <<
         \relative c'' \voiceA
         \\
         \relative c'  \voiceB
       >>
     }
     \new Staff {
       \global \clef bass
       <<
         \relative c \voiceC
         \\
         \relative c \voiceD
       >>
     }
  >>
}
```



Véase también

Manual de aprendizaje: Sección “Organizing pieces with variables” in *Manual de Aprendizaje*
Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

1.6 Notación de los pentagramas



Esta sección explica cómo influir sobre la apariencia de los pentagramas, cómo imprimir partituras con más de un pentagrama y cómo añadir indicaciones de tempo y notas guía a los pentagramas.

1.6.1 Impresión de los pentagramas
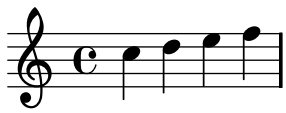
Esta sección describe los distintos métodos de creación de pentagramas y grupos de ellos.

Crear instancias de pentagramas nuevos

Las *pautas* y los *pentagramas* o pautas de cinco líneas se crean con las instrucciones `\new` o `\context`. Para ver más detalles, consulte Sección 5.1.2 [Crear contextos], página 390.

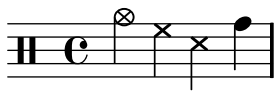
El contexto básico de pentagrama es `Staff`:

```
\new Staff { c4 d e f }
```



El contexto `DrumStaff` crea una pauta de cinco líneas preparada para un conjunto de batería típico. Cada instrumento se presenta con un símbolo distinto. Los instrumentos se escriben en el modo de percusión que sigue a una instrucción `\drummode`, con cada instrumento identificado por un nombre. Para ver más detalles, consulte [Pautas de percusión], página 253.

```
\new DrumStaff {
  \drummode { cymc hh ss tomh }
}
```



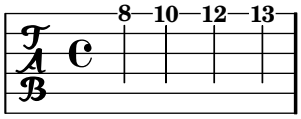
`RhythmicStaff` crea una pauta de una sola línea que sólo muestra las duraciones de la entrada. Se preservan las duraciones reales. Para ver más detalles, consulte [Mostrar los ritmos de la melodía], página 53.

```
\new RhythmicStaff { c4 d e f }
```



`TabStaff` crea una tablatura con seis cuerdas en la afinación estándar de guitarra. Para ver más detalles, consulte [Tablaturas predeterminadas], página 225.
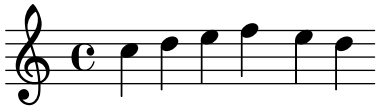
```
\new TabStaff { c4 d e f }
```



Existen dos contextos de pauta específicos para la notación de música antigua: `MensuralStaff` y `VaticanaStaff`. Se describen en [Contextos predefinidos], página 284.

`GregorianTranscriptionStaff` crea una pauta para la notación moderna de canto gregoriano. No muestra líneas divisorias.

```
\new GregorianTranscriptionStaff { c4 d e f e d }
```



Se pueden definir contextos nuevos de pentagrama único. Para ver más detalles, consulte Sección 5.1.5 [Definir contextos nuevos], página 394.

Véase también

Glosario musical: Sección “pantagrama” in *Glosario Musical*, Sección “pentagramas” in *Glosario Musical*.

Referencia de la notación: Sección 5.1.2 [Crear contextos], página 390, [Pautas de percusión], página 253, [Mostrar los ritmos de la melodía], página 53, [Tablaturas predeterminadas], página 225, [Contextos predefinidos], página 284, [El símbolo del pentagrama], página 133, [Contextos del canto gregoriano], página 294, [Contextos de la música mensural], página 286, Sección 5.1.5 [Definir contextos nuevos], página 394.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

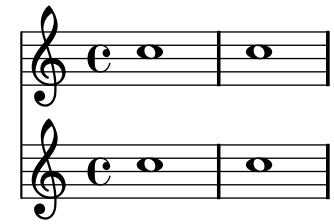
Referencia de funcionamiento interno: Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “DrumStaff” in *Referencia de Funcionamiento Interno*, Sección “GregorianTranscriptionStaff” in *Referencia de Funcionamiento Interno*, Sección “RhythmicStaff” in *Referencia de Funcionamiento Interno*, Sección “TabStaff” in *Referencia de Funcionamiento Interno*, Sección “MensuralStaff” in *Referencia de Funcionamiento Interno*, Sección “VaticanaStaff” in *Referencia de Funcionamiento Interno*, Sección “StaffSymbol” in *Referencia de Funcionamiento Interno*.

Agrupar pentagramas

Existen varios contextos para agrupar pentagramas individuales formando sistemas. Cada contexto de agrupación establece el estilo del delimitador de comienzo del sistema y el comportamiento de las barras de compás.

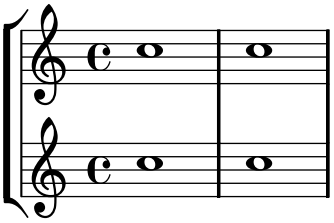
Si no se especifica ningún contexto, se usan las propiedades predeterminadas: el grupo comienza con una línea vertical y las barras de compás no están conectadas.

```
<<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



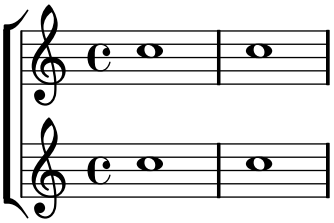
En el contexto `StaffGroup`, el grupo se inicia con un corchete y las barras de compás se dibujan atravesando todos los pentagramas.

```
\new StaffGroup <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



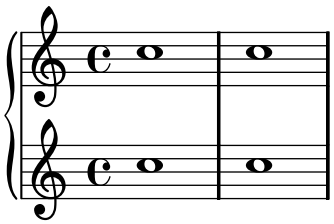
En un `ChoirStaff` (sistema de coro), el grupo se inicia con un corchete, pero las barras de compás no están conectadas.

```
\new ChoirStaff <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



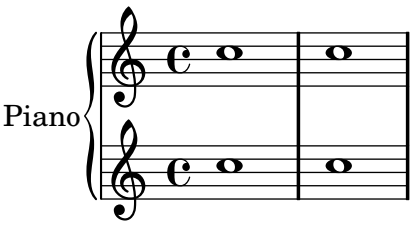
En un `GrandStaff` (sistema de piano), el grupo se inicia con una llave y las barras de compás se conectan entre los pentagramas.

```
\new GrandStaff <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



El `PianoStaff` (sistema de piano) es idéntico a `GrandStaff`, excepto que contempla directamente la impresión del nombre del instrumento. Para ver más detalles, consulte [Nombres de instrumentos], página 145.

```
\new PianoStaff <<
  \set PianoStaff.instrumentName = #"Piano"
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



Cada contexto de grupo de pentagramas fija la propiedad del delimitador de inicio `systemStartDelimiter` a uno de los siguientes valores: `SystemStartBar` (línea), `SystemStartBrace` (llave) o `SystemStartBracket` (corchete). También está disponible un cuarto delimitador, `SystemStartSquare` (corchete en ángulo recto), pero se debe especificar explícitamente.

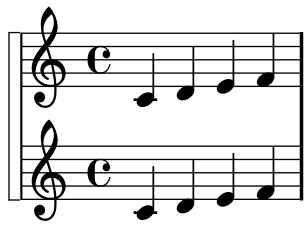
Se pueden definir contextos nuevos de grupo de pentagramas. Para ver más detalles, consulte Sección 5.1.5 [Definir contextos nuevos], página 394.

Fragmentos de código seleccionados

Uso del corchete recto al comienzo de un grupo de pentagramas

Se puede usar el delimitador de comienzo de un sistema `SystemStartSquare` estableciéndolo explícitamente dentro de un contexto `StaffGroup` o `ChoirStaffGroup`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



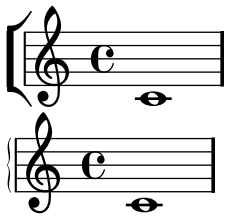
Mostrar corchete o llave en grupos de un solo pentagrama

Si hay un solo pentagrama en un de los tipos de sistema `ChoirStaff` o `StaffGroup`, el comportamiento predeterminado es que no se imprima el corchete en la barra inicial. Esto se puede cambiar sobreescribiendo las propiedades adecuadas.

Observe que en contextos como `PianoStaff` y `GrandStaff` en que los sistemas empiezan con una llave en lugar de un corchete, se debe establecer el valor de una propiedad distinta, como se ve en el segundo sistema del ejemplo.

```
\markup \left-column {
  \score {
    \new StaffGroup <<
```

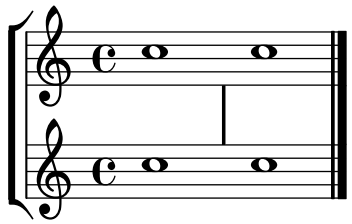
```
      % debe ser menor que el número real de líneas de la pauta
      \override StaffGroup.SystemStartBracket #'collapse-height = #1
      \override Score.SystemStartBar #'collapse-height = #1
      \new Staff {
        c'1
      }
    >>
    \layout { }
  }
  \score {
    \new PianoStaff <<
      \override PianoStaff.SystemStartBrace #'collapse-height = #1
      \override Score.SystemStartBar #'collapse-height = #1
      \new Staff {
        c'1
      }
    >>
    \layout { }
  }
}
```



Disposición Mensurstriche (líneas divisorias entre pentagramas)

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir con un `StaffGroup` en vez de un `ChoirStaff`. La línea divisoria sobre los pentagramas se borra estableciendo la propiedad `transparent`.

```
global = {
  \override Staff.BarLine #'transparent = ##t
  s1 s
  % la barra final no se interrumpe
  \revert Staff.BarLine #'transparent
  \bar "|."
}
\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



Véase también

Glosario musical: Sección “llave” in *Glosario Musical*, Sección “llave o corchete” in *Glosario Musical*, Sección “sistema de piano” in *Glosario Musical*.

Referencia de la notación: [Nombres de instrumentos], página 145, Sección 5.1.5 [Definir contextos nuevos], página 394.

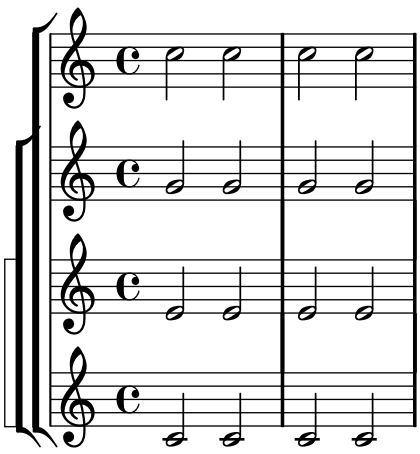
Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “StaffGroup” in *Referencia de Funcionamiento Interno*, Sección “ChoirStaff” in *Referencia de Funcionamiento Interno*, Sección “GrandStaff” in *Referencia de Funcionamiento Interno*, Sección “PianoStaff” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBar” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBrace” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBracket” in *Referencia de Funcionamiento Interno*, Sección “SystemStartSquare” in *Referencia de Funcionamiento Interno*.

Grupos de pentagramas anidados

Los contextos de grupos de pentagramas se pueden anidar hasta una profundidad arbitraria. En este caso, cada contexto descendiente crea un corchete nuevo adyacente al corchete de su grupo padre.

```
\new StaffGroup <<
  \new Staff { c2 c | c2 c }
  \new StaffGroup <<
    \new Staff { g2 g | g2 g }
    \new StaffGroup \with {
      systemStartDelimiter = #'SystemStartSquare
    }
    <<
      \new Staff { e2 e | e2 e }
      \new Staff { c2 c | c2 c }
    >>
  >>
>>
```



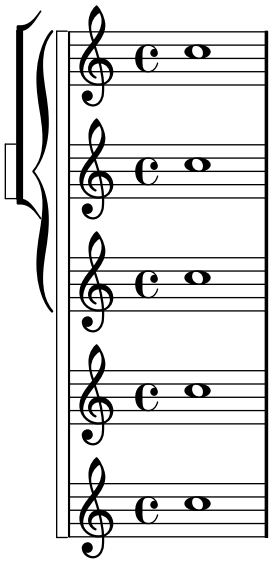
Se pueden definir nuevos contextos de grupos de pentagramas anidados. Para ver más detalles, consulte Sección 5.1.5 [Definir contextos nuevos], página 394.

Fragmentos de código seleccionados

Anidado de grupos de pentagramas

Se puede utilizar la propiedad `systemStartDelimiterHierarchy` para crear grupos de pentagramas anidados de forma más compleja. La instrucción `\set StaffGroup.systemStartDelimiterHierarchy` toma una lista alfabética del número de pentagramas producidos. Se puede proporcionar antes de cada pentagrama un delimitador de comienzo de sistema. Se debe encerrar entre corchetes y admite tantos pentagramas como encierren las llaves. Se pueden omitir los elementos de la lista, pero el primer corchete siempre abarca todos los pentagramas. Las posibilidades son `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` y `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
                              (SystemStartSquare b)  ) c ) d)
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Véase también

Referencia de la notación: [Agrupar pentagramas], página 127, [Nombres de instrumentos], página 145, Sección 5.1.5 [Definir contextos nuevos], página 394.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffGroup” in *Referencia de Funcionamiento Interno*, Sección “ChoirStaff” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBar” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBrace” in *Referencia de Funcionamiento Interno*, Sección “SystemStartBracket” in *Referencia de Funcionamiento Interno*, Sección “SystemStartSquare” in *Referencia de Funcionamiento Interno*.

1.6.2 Modificación de pentagramas sueltos
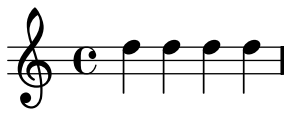
Esta sección explica cómo cambiar los atributos específicos de un pentagrama: por ejemplo, cambiar el número de líneas o el tamaño de la pauta. También se describen los métodos para comenzar y terminar los pentagramas, y establecer secciones de ossia.

El símbolo del pentagrama

las líneas de un pentagrama pertenecen al grob `StaffSymbol`. Se pueden modificar las propiedades de `StaffSymbol` para modificar la apariencia de una pauta, pero han de modificarse antes de que se produzca la creación de la misma.
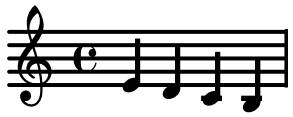
Se puede cambiar el número de líneas de la pauta. La posición de la clave y del Do central pueden requerir una modificación para mantener la correspondencia con la nueva pauta. Para ver una explicación, consulte la sección de fragmentos de código en [Clave], página 12.

```
\new Staff \with {
  \override StaffSymbol #'line-count = #3
}
{ d4 d d d }
```
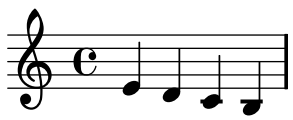


Se puede modificar el grosor de las líneas de la pauta. El grosor de las líneas adicionales y las plicas también resultan afectados, pues dependen del grosor de las líneas de la pauta.

```
\new Staff \with {
  \override StaffSymbol #'thickness = #3
}
{ e4 d c b }
```



El grosor de las líneas adicionales se puede fijar de forma independiente al de las líneas de la pauta. En el ejemplo, los dos números son factores que multiplican el grosor de las líneas del pentagrama y su espaciado. Las dos contribuciones se suman para dar el grosor de las líneas adicionales.

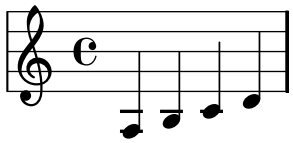
```
\new Staff \with {
  \override StaffSymbol #'ledger-line-thickness = #'(1 . 0.2)
}
{ e4 d c b }
```



Se puede cambiar la distancia entre las líneas de la pauta. Este ajuste afecta también al espaciado de las líneas adicionales.

```
\new Staff \with {
  \override StaffSymbol #'staff-space = #1.5
}
```

```
{ a4 b c d }
```



Hay más detalles sobre las propiedades de `StaffSymbol` en Sección “staff-symbol-interface” in *Referencia de Funcionamiento Interno*.

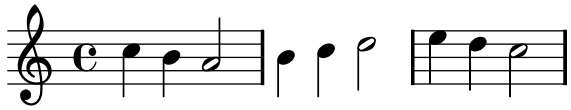
Se pueden colocar modificaciones a las propiedades de una pauta en el medio de una partitura, entre `\stopStaff` (detener pauta) y `\startStaff` (iniciar pauta):

```
c2 c
\stopStaff
\override Staff.StaffSymbol #'line-count = #2
\startStaff
b2 b
\stopStaff
\revert Staff.StaffSymbol #'line-count
\startStaff
a2 a
```



En general, se pueden usar `\startStaff` y `\stopStaff` para detener o iniciar un pentagrama en mitad de una partitura.

```
c4 b a2
\stopStaff
b4 c d2
\startStaff
e4 d c2
```



Instrucciones predefinidas
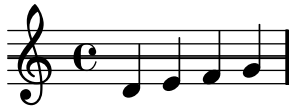
`\startStaff`, `\stopStaff`.

Fragmentos de código seleccionados

Hacer unas líneas del pentagrama más gruesas que las otras

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol #'line-positions = #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Véase también

Glosario musical: Sección “línea” in *Glosario Musical*, Sección “línea adicional” in *Glosario Musical*, Sección “pantagrama” in *Glosario Musical*.

Referencia de la notación: [Clave], página 12.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffSymbol” in *Referencia de Funcionamiento Interno*, Sección “staff-symbol-interface” in *Referencia de Funcionamiento Interno*.
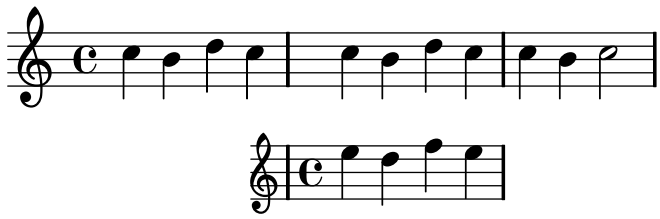
Advertencias y problemas conocidos

Al establecer manualmente las posiciones de las líneas de la pauta, las barras de compás siempre se centran alrededor de la posición 0, por lo que la distancia máxima entre las barras de compás más externas en cualquiera de las direcciones, debe ser igual.

Pentagramas de Ossia

Los pentagramas *Ossia* se pueden preparar mediante la creación de un pentagrama simultáneo nuevo en la posición adecuada:

```
\new Staff \relative c'' {
  c4 b d c
  <<
    { c4 b d c }
    \new Staff { e4 d f e }
  >>
  c4 b c2
}
```



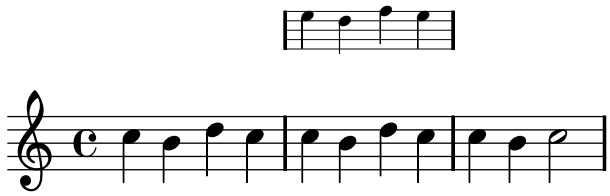
Sin embargo, el ejemplo anterior no es lo que normalmente se desea. Para crear pentagramas de ossia que estén encima del pentagrama original, que no tengan compás ni clave, y que tengan un tamaño menor de tipografía se deben usar varios trucos. El Manual de aprendizaje describe una técnica específica para llegar a este objetivo, empezando por Sección “Nesting music expressions” in *Manual de Aprendizaje*.

El ejemplo siguiente utiliza la propiedad `alignAboveContext` para alinear el pentagrama de ossia. Este método es muy conveniente cuando se necesitan sólo algunos pentagramas de ossia.

```
\new Staff = principal \relative c'' {
  c4 b d c
  <<
    { c4 b d c }

    \new Staff \with {
      \remove "Time_signature_engraver"
      alignAboveContext = #"principal"
```
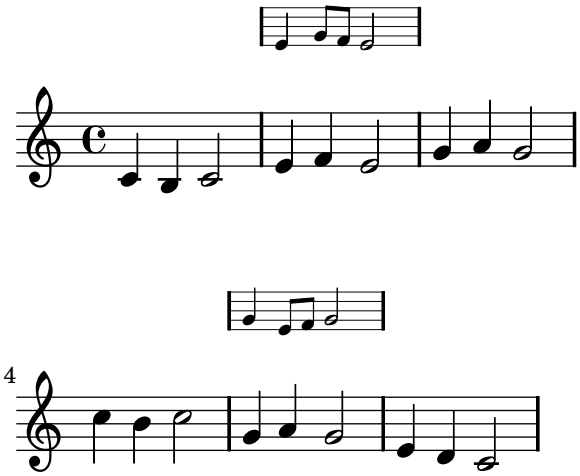
```
      fontSize = #-3
      \override StaffSymbol #'staff-space = #(magstep -3)
      \override StaffSymbol #'thickness = #(magstep -3)
      firstClef = ##f
    }
    { e4 d f e }
  >>
  c4 b c2
}
```



Si se requieren muchos pentagramas de ossia aislados, puede ser más conveniente la creación de un contexto `Staff` vacío con un *identificador de contexto* específico; después se pueden crear los pentagramas de ossia *llamando* a este contexto y usando `\startStaff` y `\stopStaff` en los puntos deseados. Las ventajas de este método son más patentes si la pieza es más larga que en ejemplo siguiente.

```
<<
  \new Staff = ossia \with {
    \remove "Time_signature_engraver"
    \override Clef #'transparent = ##t
    fontSize = #-3
    \override StaffSymbol #'staff-space = #(magstep -3)
    \override StaffSymbol #'thickness = #(magstep -3)
  }
  { \stopStaff s1*6 }
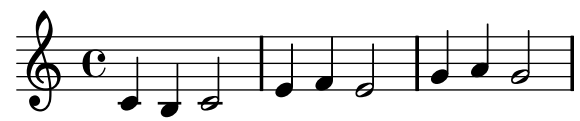
  \new Staff \relative c' {
    c4 b c2
    <<
      { e4 f e2 }
      \context Staff = ossia {
        \startStaff e4 g8 f e2 \stopStaff
      }
    >>
    g4 a g2 \break
    c4 b c2
    <<
      { g4 a g2 }
      \context Staff = ossia {
        \startStaff g4 e8 f g2 \stopStaff
      }
    >>
    e4 d c2
  }
>>
```
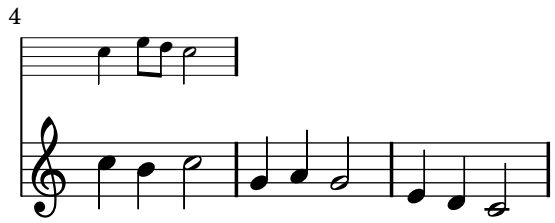


Como alternativa, se puede usar la instrucción `\RemoveEmptyStaffContext` para crear pentagramas de ossia. Este método es muy conveniente cuando los pentagramas de ossia aparecen inmediatamente después de un salto de línea. En este caso, no es necesario en absoluto utilizar silencios de separación; solamente son necesarios `\startStaff` y `\stopStaff`. Para ver más información sobre `\RemoveEmptyStaffContext`, consulte [Ocultar pentagramas], página 139.

```
<<
  \new Staff = ossia \with {
    \remove "Time_signature_engraver"
    \override Clef #'transparent = ##t
    fontSize = #-3
    \override StaffSymbol #'staff-space = #(magstep -3)
    \override StaffSymbol #'thickness = #(magstep -3)
  }
  \new Staff \relative c' {
    c4 b c2
    e4 f e2
    g4 a g2 \break
    <<
      { c4 b c2 }
      \context Staff = ossia {
        c4 e8 d c2 \stopStaff
      }
    >>
    g4 a g2
    e4 d c2
  }
>>

\layout {
  \context {
    \RemoveEmptyStaffContext
    \override VerticalAxisGroup #'remove-first = ##t
  }
}
```





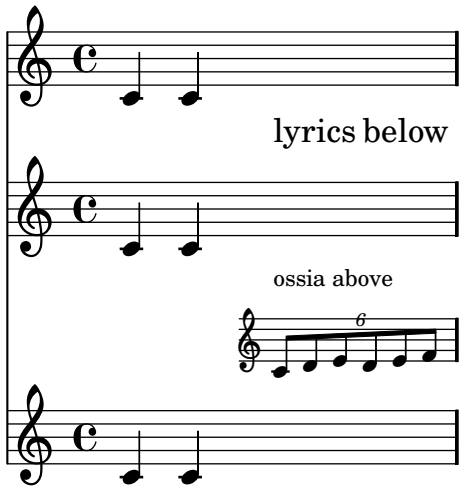
Fragmentos de código seleccionados

Alineación vertical de la letra y los compases de ossia

Este fragmento de código muestra el uso de las propiedades de contexto `alignBelowContext` y `alignAboveContext` para controlar la posición de la letra y los compases de ossia.

```
\paper {
  ragged-right = ##t
}

\relative c' <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = #"3"
        fontSize = #-2
        \override StaffSymbol #'staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
      } {
        \times 4/6 {
          \override TextScript #'padding = #3
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



Véase también

Glosario musical: Sección “ossia” in *Glosario Musical*, Sección “pantagrama” in *Glosario Musical*, Sección “Sistemas a la francesa” in *Glosario Musical*.

Mannual de aprendizaje: Sección “Nesting music expressions” in *Manual de Aprendizaje*, Sección “Size of objects” in *Manual de Aprendizaje*, Sección “Length and thickness of objects” in *Manual de Aprendizaje*.

Referencia de la notación: [Ocultar pentagramas], página 139.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffSymbol” in *Referencia de Funcionamiento Interno*.

Ocultar pentagramas

Se pueden ocultar las líneas del pentagrama quitando el grabador `Staff_symbol_engraver` del contexto de `Staff`. Como alternativa se puede utilizar la instrucción `\stopStaff`.

```
\new Staff \with {
  \remove "Staff_symbol_engraver"
}
\relative c''' { a8 f e16 d c b a2 }
```

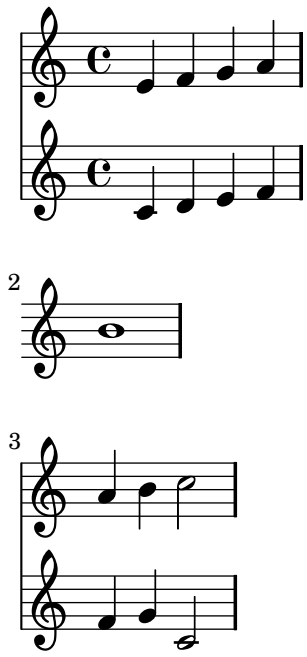


Se pueden ocultar los pentagramas que están vacíos mediante la instrucción `\RemoveEmptyStaffContext` dentro del bloque `\layout`. En las partituras orquestales, este estilo se conoce como ‘partitura a la francesa’. De forma predeterminada, esta instrucción oculta y elimina todos los pentagramas vacíos de la partitura excepto los del primer sistema.

Nota: Un pentagrama se considera vacío cuando contiene solamente silencios multicompás, desplazamientos, silencios de separación o una combinación de estos elementos.

```
\layout {
  \context {
    \RemoveEmptyStaffContext
  }
}
```

```
\relative c' <<
  \new Staff {
    e4 f g a \break
    b1 \break
    a4 b c2
  }
  \new Staff {
    c,4 d e f \break
    R1 \break
    f4 g c,2
  }
>>
```



También se puede usar `\RemoveEmptyStaffContext` para crear secciones de ossia para un pentagrama. Para ver más detalles, consulte [Pentagramas de Ossia], página 135.

Se puede usar la instrucción `\AncientRemoveEmptyStaffContext` para ocultar los pentagramas vacíos en contextos de música antigua. de forma similar, se puede usar `\RemoveEmptyRhythmicStaffContext` para ocultar los contextos `RhythmicStaff` vacíos.

Instrucciones predefinidas

`\RemoveEmptyStaffContext`, `\AncientRemoveEmptyStaffContext`, `\RemoveEmptyRhythmicStaffContext`.

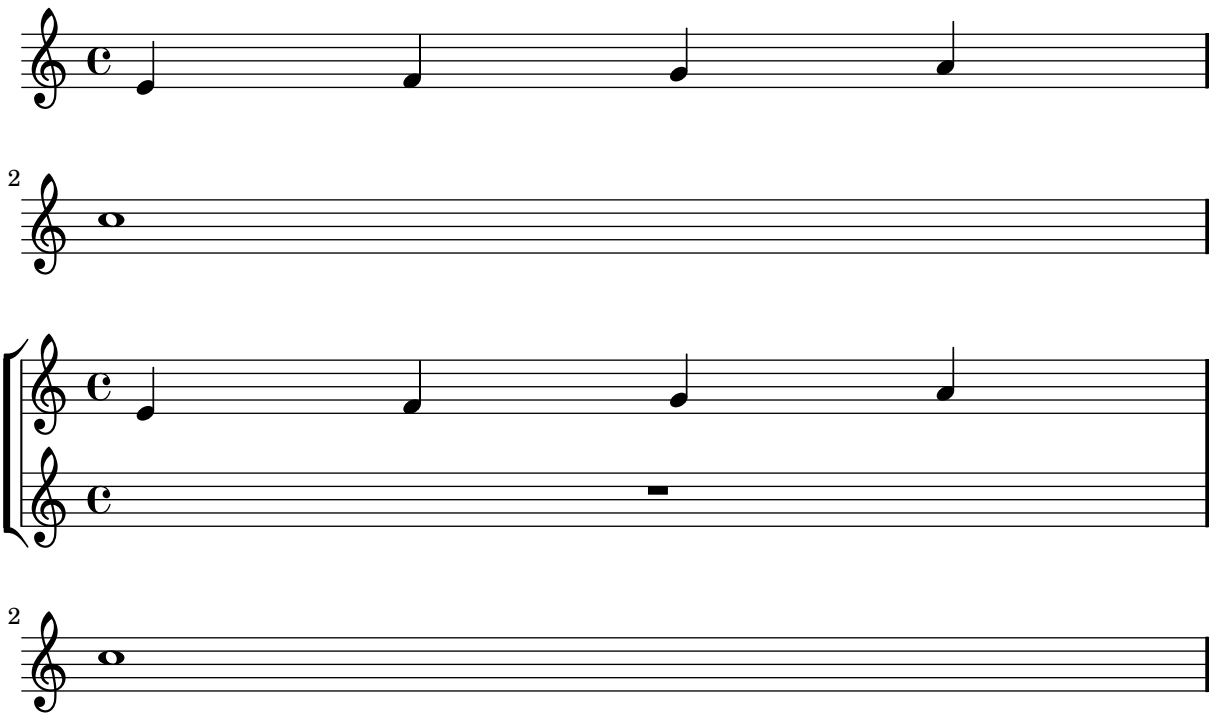
Fragmentos de código seleccionados

Quitar la primera línea vacía

El primer pentagrama vacío también se puede suprimir de la partitura estableciendo la propiedad `remove-first` de `VerticalAxisGroup`. Esto se puede hacer globalmente dentro del bloque `\layout`, o localmente dentro del pentagrama concreto que se quiere suprimir. En este último caso, tenemos que especificar el contexto (`Staff` se aplica sólo al pentagrama actual) delante de la propiedad.

El pentagrama inferior del segundo grupo no se elimina, porque el ajuste sólo se aplica al pentagrama concreto dentro del que se escribe.

```
\layout {
  \context {
    \RemoveEmptyStaffContext
    % Para usar el ajuste de forma global, quite el comentario de la líne siguiente:
    % \override VerticalAxisGroup #'remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % Para usar el ajuste globalmente, haga un comentario de la línea siguiente:
    % quite el comentario de la línea del bloque \layout de arriba
    \override Staff.VerticalAxisGroup #'remove-first = ##t
    R1 \break
    R
  }
>>
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
>>
```



Véase también

Glosario musical: Sección “Sistemas a la francesa” in *Glosario Musical*.

Referencia de la notación: [El símbolo del pentagrama], página 133, [Pentagramas de Ossia], página 135.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ChordNames” in *Referencia de Funcionamiento Interno*, Sección “FiguredBass” in *Referencia de Funcionamiento Interno*, Sección “Lyrics” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*, Sección “VerticalAxisGroup” in *Referencia de Funcionamiento Interno*, Sección “Staff_symbol_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La eliminación del grabador `Staff_symbol_engraver` también oculta las barras de compás. Si se fuerza la visibilidad de la barra de compás, pueden ocurrir errores de formato visual. En este caso, utilice las siguientes sobreescrituras de valores en vez de quitar el grabador:

```
\override StaffSymbol #'stencil = ##f
\override NoteHead #'no-ledgers = ##t
```
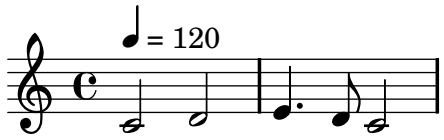
1.6.3 Escritura de las particellas

Esta sección explica cómo insertar indicaciones de tempo y nombres de instrumento en una partitura. También se describen métodos para citar otras voces y dar formato a las notas guía.

Indicaciones metronómicas
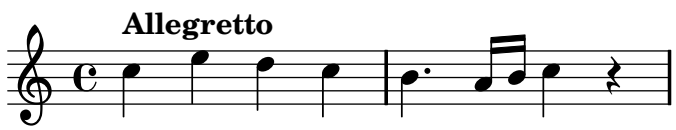
Es muy sencillo escribir una indicación metronómica básica:

```
\tempo 4 = 120
c2 d
e4. d8 c2
```
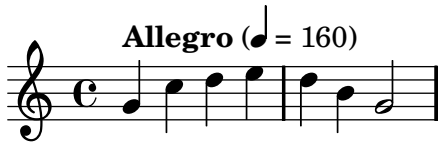


Se pueden usar indicaciones de tempo con texto:

```
\tempo "Allegretto"
c4 e d c
b4. a16 b c4 r4
```
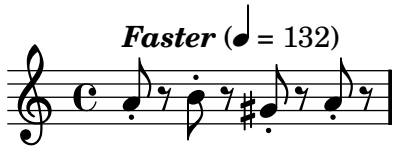


La combinación de una indicación metronómica y un texto hace que la marca de metrónomo se encierre entre paréntesis automáticamente:

```
\tempo "Allegro" 4 = 160
g4 c d e
d4 b g2
```
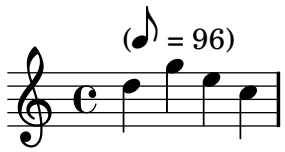


En general, el texto puede ser cualquier objeto de marcado:

```
\tempo \markup { \italic Faster } 4 = 132
a8-. r8 b-. r gis-. r a-. r
```



Se puede escribir una indicación metronómica entre paréntesis sin ninguna indicación textual, escribiendo una cadena vacía en la entrada:

```
\tempo "" 8 = 96
d4 g e c
```



Fragmentos de código seleccionados

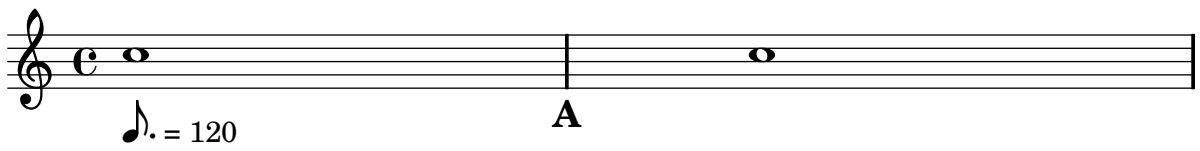
Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama

De forma predeterminada, las indicaciones metronómicas y las letras de ensayo se imprimen encima del pentagrama. Para colocarlas debajo del pentagrama, simplemente ajustamos adecuadamente las propiedades `side-axis` y `direction` de `MetronomeMark` o de `RehearsalMark`.

```
\layout { ragged-right = ##f }

{
  % las marcas de metrónomo debajo del pentagrama
  \override Score.MetronomeMark #'direction = #DOWN
  \tempo 8. = 120
  c''1
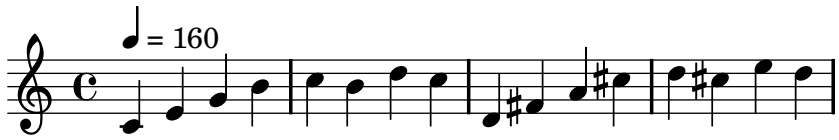
  % Letras de ensayo debajo del pentagrama
  \override Score.RehearsalMark #'direction = #DOWN
  \mark \default
  c''1
}
```



Cambiar el tempo sin indicación metronómica

Para cambiar el tempo en la salida MIDI sin imprimir nada, hacemos invisible la indicación metronómica:
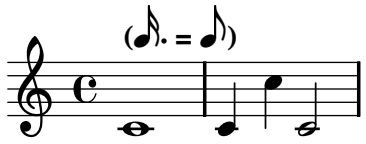
```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



Crear indicaciones metronómicas en modo de marcado

Se pueden crear indicaciones metronómicas nuevas en modo de marcado, pero no cambian el tempo en la salida MIDI.

```
\relative c' {
  \tempo \markup {
    \concat {
      (
      \smaller \general-align #Y #DOWN \note #"16." #1
      " = "
      \smaller \general-align #Y #DOWN \note #"8" #1
      )
    }
  }
  c1
  c4 c' c,2
}
```



Para ver más detalles, consulte Sección 1.8.2 [Formatear el texto], página 173.

Véase también

Glosario musical: Sección “metrónomo” in *Glosario Musical*, Sección “indicación metronómica” in *Glosario Musical*, Sección “indicación del tempo” in *Glosario Musical*, Sección “indicación metronómica” in *Glosario Musical*.

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 173, Sección 3.5 [Salida MIDI], página 335.
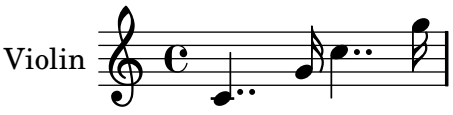
Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.
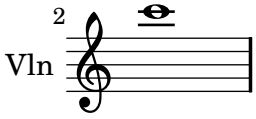
Referencia de funcionamiento interno:: Sección “MetronomeMark” in *Referencia de Funcionamiento Interno*.

Nombres de instrumentos

Se pueden imprimir los nombres de los instrumentos en el lado izquierdo de los pentagramas dentro de los contextos `Staff` y `PianoStaff`. El valor de `instrumentName` se usa para el primer pentagrama, y el valor de `shortInstrumentName` se usa para todos los pentagramas siguientes.
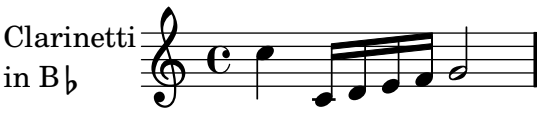
```
\set Staff.instrumentName = #"Violin "
\set Staff.shortInstrumentName = #"Vln "
c4.. g'16 c4.. g'16
\break
c1
```
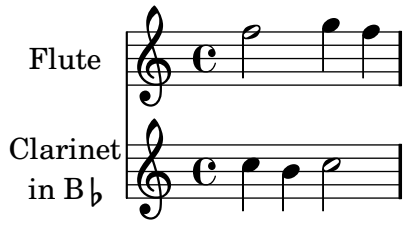




También podemos usar el modo de marcado para construir nombres de instrumento más complicados:

```
\set Staff.instrumentName = \markup {
  \column { "Clarinetti"
            \line { "in B" \smaller \flat } } }
c4 c,16 d e f g2
```



Cuando se agrupan dos o más contextos de pentagrama, los nombres de instrumento y los nombres cortos aparecen centrados de forma predeterminada. Para centrar nombres de instrumento de varias líneas, se debe utilizar `\center-column`:

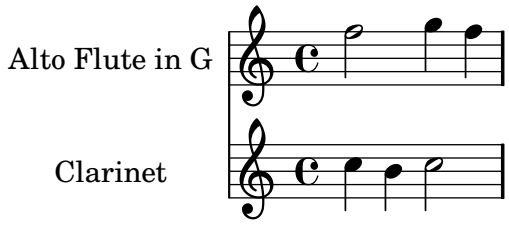
```
<<
  \new Staff {
    \set Staff.instrumentName = #"Flute"
    f2 g4 f
  }
  \new Staff {
    \set Staff.instrumentName = \markup \center-column {
      Clarinet
      \line { "in B" \smaller \flat }
    }
    c4 b c2
  }
>>
```
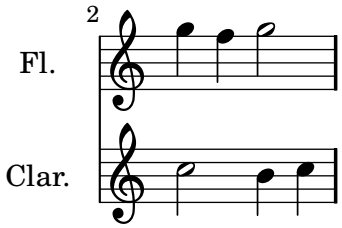


Sin embargo, si los nombres de instrumento son más largos, éstos no se centran para un grupo de pentagramas a no ser que se aumenten los valores del sangrado, `indent`, y del sangrado corto, `short-indent`. Para ver más detalles sobre estos ajustes, consulte [Dimensiones horizontales], página 347.

```
\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
}

\relative c'' <<
  \new Staff {
    \set Staff.instrumentName = #"Alto Flute in G"
    \set Staff.shortInstrumentName = #"Fl."
    f2 g4 f \break
    g4 f g2
  }
  \new Staff {
    \set Staff.instrumentName = #"Clarinet"
    \set Staff.shortInstrumentName = #"Clar."
    c,4 b c2 \break
    c2 b4 c
  }
>>
```



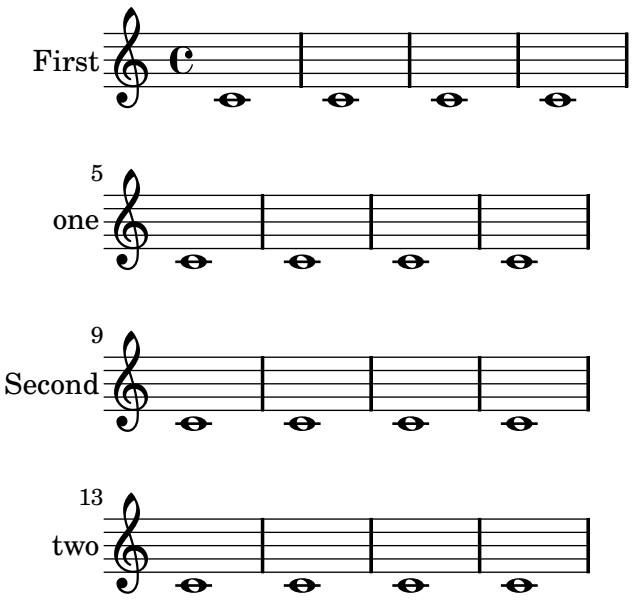


Para añadir nombres de instrumento a otros contextos (como `GrandStaff`, `ChoirStaff` o `StaffGroup`), debemos añadir el grabador `Instrument_name_engraver` a dicho contexto. Para ver más detalles, consulte Sección 5.1.3 [Modificar los complementos (plug-ins) de contexto], página 391.

Los nombres de instrumento se pueden cambiar en mitad de una pieza:

```
\set Staff.instrumentName = #"First"
\set Staff.shortInstrumentName = #"one"
```
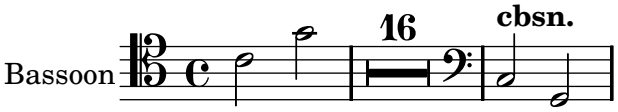
```
c1 c c c \break
c1 c c c \break
\set Staff.instrumentName = #"Second"
\set Staff.shortInstrumentName = #"two"
c1 c c c \break
c1 c c c \break
```

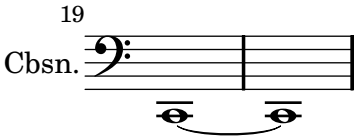


se se necesita un *cambio* de instrumento, se debe usar `\addInstrumentDefinition` en combinación con `\instrumentSwitch` para crear una lista detallada de los cambios necesarios para el intercambio. La instrucción `\addInstrumentDefinition` tiene dos argumentos: una cadena identificativa, y una lista asociativa de propiedades de contexto y los valores que el instrumento va a usar. Se debe situar en el ámbito del nivel más alto. `\instrumentSwitch` se usa en la expresión musical para declarar el cambio de instrumento:

```
\addInstrumentDefinition #"contrabassoon"
  #`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
     (shortInstrumentName . "Cbsn.")
     (clefGlyph . "clefs.F")
     (middleCPosition . 6)
     (clefPosition . 2)
     (instrumentCueName . ,(make-bold-markup "cbsn."))
     (midiInstrument . "bassoon"))

\new Staff \with {
  instrumentName = #"Bassoon"
}
\relative c' {
  \clef tenor
  \compressFullBarRests
  c2 g'
  R1*16
  \instrumentSwitch "contrabassoon"
  c,,2 g \break
  c,1 ~ | c1
}
```





Véase también

Referencia de la notación: [Dimensiones horizontales], página 347, Sección 5.1.3 [Modificar los complementos (plug-ins) de contexto], página 391.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “InstrumentName” in *Referencia de Funcionamiento Interno*, Sección “PianoStaff” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*.

Citar otras voces

Es muy frecuente que una voz duplique música de otra voz. Por ejemplo, los violines primero y segundo pueden tocar las mismas notas durante un pasaje musical. En LilyPond esto se hace dejando que una voz *cite* a la otra, sin tener que volverla a introducir.
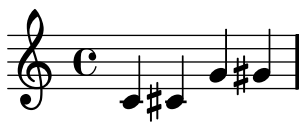
Antes de que una voz pueda citarse, se debe usar la instrucción `\addQuote` para iniciar el fragmento citado. Esta instrucción se debe usar en el ámbito del nivel más alto. El primer argumento es una cadena identificativa, y el segundo es una expresión musical:

```
flute = \relative c'' {
  a4 gis g gis
}
\addQuote "flute" { \flute }
```

La instrucción `\quoteDuring` se usa para indicar el punto en que comienza la cita. va seguido por dos argumentos: el nombre de la voz citada, tal y como se definió con `\addQuote`, y una expresión musical que indica la duración de la cita, normalmente silencios de separación o silencios multicompás. La música correspondiente de la voz citada se inserta dentro de la expresión musical:

```
flauta = \relative c'' {
  a4 gis g gis
}
\addQuote "flute" { \flauta }
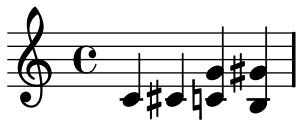
\relative c' {
  c4 cis \quoteDuring #"flute" { s2 }
}
```



Si la expresión musical que se usa para `\quoteDuring` contiene cualquier cosa que no sea un silencio de separación o un silencio multicompás, ase produce una situación de polifonía, lo que no suele ser deseable:

```
flauta = \relative c'' {
  a4 gis g gis
}
\addQuote "flute" { \flauta }
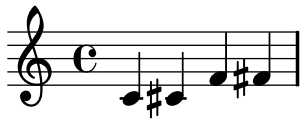
\relative c' {
  c4 cis \quoteDuring #"flute" { c4 b }
}
```



Las citas reconocen los ajustes de transposición de los instrumentos tanto para la fuente como para los instrumentos de destino si se utiliza la instrucción `\transposition`. Para ver más detalles sobre `\transposition`, consulte [Transposición de los instrumentos], página 17.

```
clarinete = \relative c'' {
  \transposition bes
  a4 gis g gis
}
\addQuote "clarinet" { \clarinete }

\relative c' {
  c4 cis \quoteDuring #"clarinet" { s2 }
}
```



Es posible etiquetar las citas con nombres exclusivos para poder procesarlas de diferentes maneras. Para ver más detalles sobre este procedimiento, consulte [Uso de etiquetas], página 330.

Fragmentos de código seleccionados

Citar otra voz con transposición

Los pasajes citados tienen en cuenta la transposición de la fuente tanto como la del destino. En este ejemplo, todos los instrumentos interpreta una nota con el sonido del Do central; el destino de un instrumento transpositor en Fa. La parte de destino se puede transponer utilizando `\transpose`. En este caso se transportan todas las notas (incluidas las citadas).
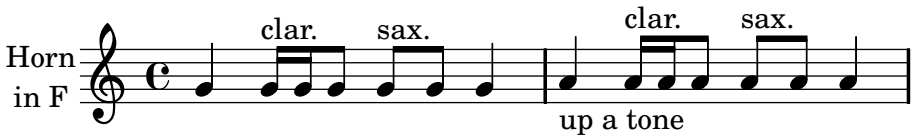
```
\addQuote clarinet {
  \transposition bes
  \repeat unfold 8 { d'16 d' d'8 }
}

\addQuote sax {
  \transposition es'
  \repeat unfold 16 { a8 }
}

pruebaDeCita = {
  % trompa
```

```
  \transposition f
  g'4
  << \quoteDuring #"clarinet" { \skip 4 } s4^"clar." >>
  << \quoteDuring #"sax" { \skip 4 } s4^"sax." >>
  g'4
}

{
  \set Staff.instrumentName =
    \markup {
      \center-column { Horn \line { in F } }
    }
  \pruebaDeCita
  \transpose c' d' << \pruebaDeCita s4_"up a tone" >>
}
```



Citar otra voz
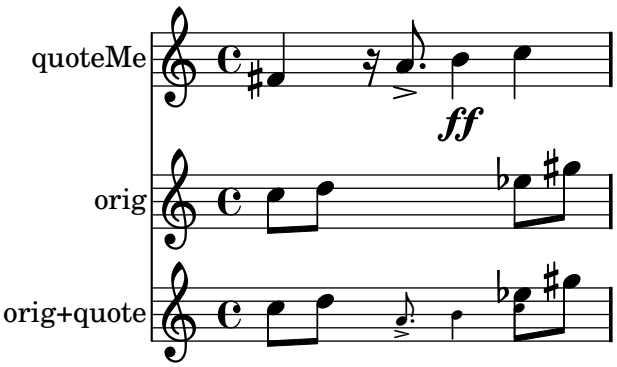
La propiedad `quotedEventTypes` determina los tipos de eventos musicales que resultan citados. El valor predeterminado es `(note-event rest-event)`, que significa que sólo aparecen en la expresión `\quoteDuring` las notas y los silencios. En el ejemplo siguiente, el silencio de semicorchea no aparece en el fragmento citado porque `rest-event` no está dentro de los `quotedEventTypes`.

```
citaMe = \relative c' {
  fis4 r16 a8.-> b4\ff c
}
\addQuote citaMe \citaMe

original = \relative c'' {
  c8 d s2
  \once \override NoteColumn #'ignore-collision = ##t
  es8 gis8
}

<<
  \new Staff {
    \set Staff.instrumentName = #"quoteMe"
    \citaMe
  }
  \new Staff {
    \set Staff.instrumentName = #"orig"
    \original
  }
  \new Staff \relative c'' <<
    \set Staff.instrumentName = #"orig+quote"
    \set Staff.quotedEventTypes =
      #'(note-event articulation-event)
    \original
```

```
    \new Voice {
      s4
      \set fontSize = #-4
      \override Stem #'length-fraction = #(magstep -4)
      \quoteDuring #"quoteMe" { \skip 2. }
    }
  >>
>>
```



Véase también

Referencia de la notación: [Transposición de los instrumentos], página 17, [Uso de etiquetas], página 330.

Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “QuoteMusic” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Sólo el contenido de la primera voz de una instrucción `\addQuote` se tiene en cuenta para la cita, de manera que la música no puede contener enunciados `\new` ni `\context Voice` que cambiarían a una voz distinta.

La cita de notas de adorno no es funcional e incluso puede hacer que LilyPond termine de forma abrupta.

La cita de unos tresillos dentro de otros puede dar como resultado una notación de pobre calidad.

En anteriores versiones LilyPond (antes de la 2.11), `addQuote` se escribía completamente en minúsculas: `\addquote`.

Formateo de las notas de aviso
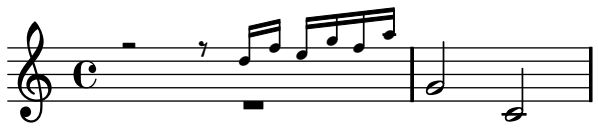
La sección anterior explica la forma de crear citas. La instrucción `\cueDuring` es una forma más especializada de `\quoteDuring`, que es especialmente útil para insertar notas guía en una particella. La sintaxis es como sigue:

`\cueDuring #`*`nombre_de_la_parte`* `#`*`voz música`*

Esta instrucción copa los compases correspondientes a partir de *nombre_de_la_parte* dentro de un contexto `CueVoice` de voz guía. La voz guía `CueVoice` se crea implícitamente, y se produce simultáneamente con *música*, lo que crea una situación polifónica. El argumento *voz* determina si la notación de las notas guía debe ser como primera o como segunda voz; `UP` (arriba) corresponde a la primera voz, y `DOWN` (abajo) corresponde a la segunda.

```
oboe = \relative c'' {
  r2 r8 d16 f e g f a
  g8 g16 g g2.
}
\addQuote "oboe" { \oboe }

\new Voice \relative c'' {
  \cueDuring #"oboe" #UP { R1 }
  g2 c,
}
```
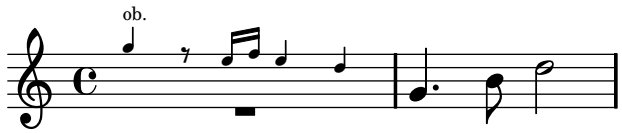


En el ejemplo anterior, el contexto `Voice` se tuvo que declarar explícitamente, pues en caso contrario toda la expresión musical habría pertenecido al contexto `CueVoice`.

Se puede imprimir el nombre del instrumento de las notas guía estableciendo la propiedad `instrumentCueName` en el contexto `CueVoice`.

```
oboe = \relative c''' {
  g4 r8 e16 f e4 d
}
\addQuote "oboe" { \oboe }

\new Staff \relative c'' <<
  \new CueVoice \with {
    instrumentCueName = "ob."
  }
  \new Voice {
    \cueDuring #"oboe" #UP { R1 }
    g4. b8 d2
  }
>>
```
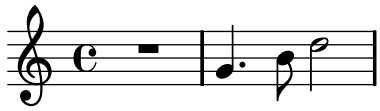


Además de imprimir el nombre del instrumento de las notas guía, cuando éstas finalizan se debe imprimir el nombre del instrumento original, y cualquier otro cambio introducido por la parte de guía se debe deshacer. Esto se puede hacer mediante la utilización de `\addInstrumentDefinition` (añadir definición de instrumento) y `\instrumentSwitch` (cambio de instrumento). Para ver un ejemplo con su explicación, consulte [Nombres de instrumentos], página 145.

La instrucción `\killCues` quita las notas guía de una expresión musical. Esto puede ser útil si las notas guía se tienen que eliminar de una particella pero posiblemente se deban reponer más tarde.

```
flauta = \relative c''' {
  r2 cis2 r2 dis2
}
```

```
\addQuote "flute" { \flauta }

\new Voice \relative c'' {
  \killCues {
    \cueDuring #"flute" #UP { R1 }
    g4. b8 d2
  }
}
```

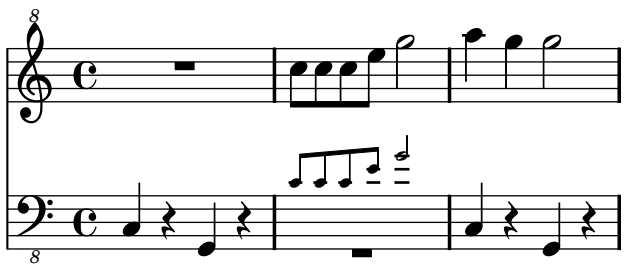


La instrucción `\transposedCueDuring` es útil para añadir guías instrumentales a partir de un registro completamente diferente. La sintaxis es similar a la de `\cueDuring`, pero necesita un argumento adicional para especificar la transposición del instrumento de guía. Para ver más información sobre la transposición, consulte [Transposición de los instrumentos], página 17.

```
flautin = \relative c''' {
  \clef "treble^8"
  R1
  c8 c c e g2
  a4 g g2
}
\addQuote "flautin" { \flautin }

cfagot = \relative c, {
  \clef "bass_8"
  c4 r g r
  \transposedCueDuring #"flautin" #UP c,, { R1 }
  c4 r g r
}

<<
  \new Staff = "flautin" \flautin
  \new Staff = "cfagot" \cfagot
>>
```



Es posible etiquetar las partes guía con numbres exclusivos de forma que se puedan procesar de diversas formas. Para ver más detalles sobre este procedimiento, consulte [Uso de etiquetas], página 330.

Véase también

Referencia de la notación: [Transposición de los instrumentos], página 17, [Nombres de instrumentos], página 145, [Uso de etiquetas], página 330.
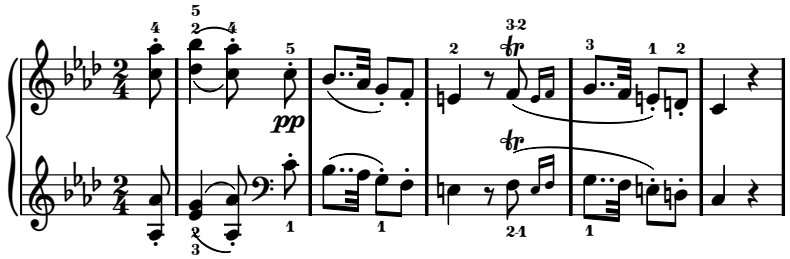
Fragmentos de código: Sección “Staff notation” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “CueVoice” in *Referencia de Funcionamiento Interno*, Sección “Voice” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Pueden ocurrir colisiones con los silencios si se usa `\cueDuring`, entre los contextos de `Voice` y de `CueVoice`.

1.7 Anotaciones editoriales



Esta sección trata de las diversas maneras de modificar el aspecto de las notas y de aplicar énfasis analítico o educativo.

1.7.1 Dentro del pentagrama

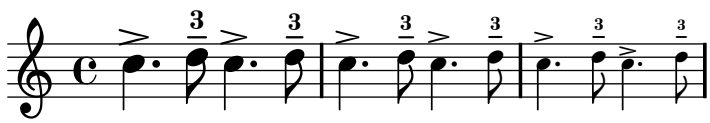
Esta sección trata sobre cómo aplicar énfasis a los elementos situados dentro del pentagrama.

Seleccionar el tamaño de la tipografía para la notación

Se puede alterar el tamaño de la fuente tipográfica de los elementos de notación. Esto no cambia el tamaño de los símbolos variables, como las barras de corchea o las ligaduras.
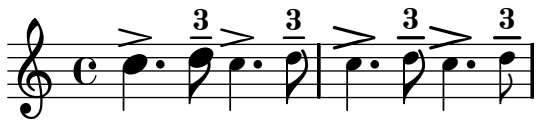
> **Nota:** Para los tamaños de tipografía del texto, consulte [Instrucciones de marcado de texto más usuales], página 175.

```
\huge
c4.-> d8---3
\large
c4.-> d8---3
\normalsize
c4.-> d8---3
\small
c4.-> d8---3
\tiny
c4.-> d8---3
\teeny
c4.-> d8---3
```



Internamente, esto establece un valor para la propiedad `fontSize`. Esto, a su vez, hace que se establezca el valor de la propiedad `font-size` de todos los objetos de presentación. El valor de `font-size` es un número que indica el tamaño con relación al tamaño estándar para la altura vigente del pentagrama. Cada unidad hacia arriba es un incremento de un 12% del tamaño de la fuente, aproximadamente. Seis unidades son exactamente un factor de dos. La función de Scheme `magstep` convierte un número de tamaño `font-size` a un factor de escalado. La propiedad `font-size` también se puede establecer directamente, de forma que queden afectados solamente determinados objetos gráficos.

```
\set fontSize = #3
c4.-> d8---3
\override NoteHead #'font-size = #-4
c4.-> d8---3
\override Script #'font-size = #2
c4.-> d8---3
\override Stem #'font-size = #-5
c4.-> d8---3
```



Los cambios en el tamaño de la fuente se obtienen por medio del escalado del tamaño del diseño que se encuentra más cerca del tamaño deseado. El tamaño estándar para la tipografía (para `font-size = #0`), depende de la altura estándar del pentagrama. Para un pentagrama de 20 puntos, se selecciona una fuente de 10 puntos.

La propiedad `font-size` sólo se puede establecer en objetos gráficos que utilicen fuentes tipográficas. Éstos son los que contemplan el interfaz de presentación `font-interface`.

Instrucciones predefinidas

`\teeny` (enano), `\tiny` (muy pequeño), `\small` (pequeño), `\normalsize` (normal), `\large` (grande), `\huge` (enorme).

Véase también
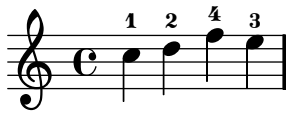
Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “font-interface” in *Referencia de Funcionamiento Interno*.

Indicaciones de digitación

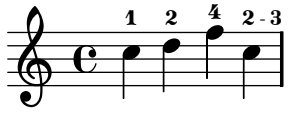
Las instrucciones de digitación se pueden introducir usando *nota-dígito*:

```
c4-1 d-2 f-4 e-3
```
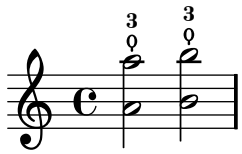


Para los cambios de dedo se pueden usar elementos de marcado de texto.

```
c4-1 d-2 f-4 c^\markup { \finger "2 - 3" }
```



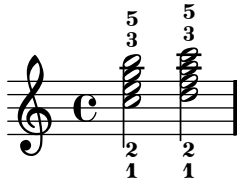
Puede usar la articulación de pulgar para indicar que una nota se debe tocar con el pulgar (p.ej. en música de cello).

```
<a_\thumb a'-3>2 <b_\thumb b'-3>
```



Las digitaciones para los acordes también se pueden añadir a las notas individuales del acorde escribiéndolas después de las alturas.

```
<c-1 e-2 g-3 b-5>2 <d-1 f-2 a-3 c-5>
```



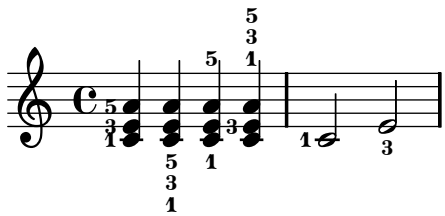
Las indicaciones de digitación se pueden situar manualmente encima o debajo del pentagrama, véase Sección 5.4.2 [Dirección y posición], página 407.

Fragmentos de código seleccionados

Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación.

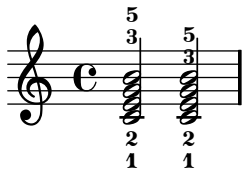
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación se imprimen de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \once \override Fingering #'staff-padding = #'()
  <c-1 e-2 g-3 b-5>2
}
```



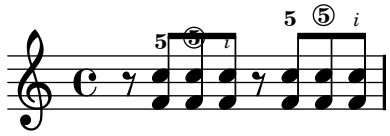
Evitar colisiones entre digitaciones de acordes y barras de corchea

Las digitaciones y los números de cuerda aplicados a notas individuales evitan a las barras automáticamente, pero de forma predeterminada esto no es cierto para las digitaciones y números de cuerda que se aplican a notas concretas de acordes. El ejemplo siguiente muestra cómo se puede sobreescribir este comportamiento predeterminado:

```
\relative c' {
  \set fingeringOrientations = #'(up)
  \set stringNumberOrientations = #'(up)
  \set strokeFingerOrientations = #'(up)

  % Comportamiento predeterminado
  r8
  <f c'-5>8
  <f c'\5>8
  <f c'-\rightHandFinger #2 >8

  % Corregido para evitar colisiones
  r8
  \override Fingering #'add-stem-support = ##t
  <f c'-5>8
  \override StringNumber #'add-stem-support = ##t
  <f c'\5>8
  \override StrokeFinger #'add-stem-support = ##t
  <f c'-\rightHandFinger #2 >8
}
```



Véase también

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 407.
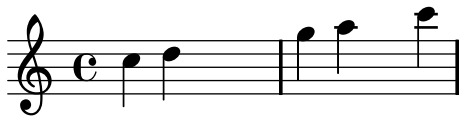
Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “FingeringEvent” in *Referencia de Funcionamiento Interno*, Sección “fingering-event” in *Referencia de Funcionamiento Interno*, Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno*, Sección “New_fingering_engraver” in *Referencia de Funcionamiento Interno*, Sección “Fingering” in *Referencia de Funcionamiento Interno*.

Notas ocultas
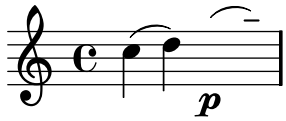
Las notas ocultas (o invisibles, o transparentes) pueden ser de utilidad en la preparación de ejercicios de teoría o de composición.

```
c4 d
\hideNotes
e4 f
\unHideNotes
g a
\hideNotes
b
\unHideNotes
c
```



Los objetos de notación que están anclados a notas invisibles, son a pesar de ello visibles.

```
c4( d)
\hideNotes
e4(\p f)--
```



Instrucciones predefinidas

`\hideNotes`, `\unHideNotes`.

Véase también

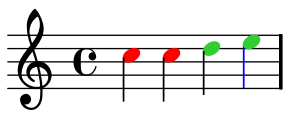
Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_spacing_engraver” in *Referencia de Funcionamiento Interno*, Sección “NoteSpacing” in *Referencia de Funcionamiento Interno*.

Colorear objetos

Se pueden asignar colores a los objetos individuales. Los nombres de color válidos se encuentran relacionados en la Sección B.5 [Lista de colores], página 460.

```
\override NoteHead #'color = #red
c4 c
\override NoteHead #'color = #(x11-color 'LimeGreen)
d
\override Stem #'color = #blue
e
```
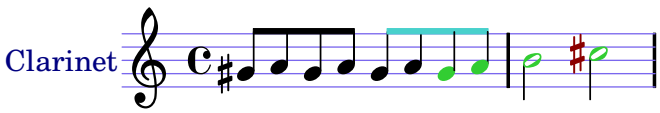


Se puede acceder al espectro completo de colores definido para X11 usando la función de Scheme x11-color. Esta función acepta un único argumento; éste puede ser un símbolo de la

forma *'FulanoMengano* o una cadena de la forma "*FulanoMengano*". La primera forma es más rápida de escribir y también más eficiente. Sin embargo, al usar la segunda forma es posible acceder a los colores de X11 por medio de la forma del nombre que tiene varias palabras.

Si `x11-color` no entiende el parámetro, el color predeterminado que se devuelve es el negro.

```
\override Staff.StaffSymbol #'color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}
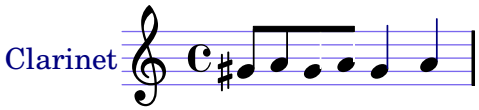
gis8 a
\override Beam #'color = #(x11-color "medium turquoise")
gis a
\override Accidental #'color = #(x11-color 'DarkRed)
gis a
\override NoteHead #'color = #(x11-color "LimeGreen")
gis a
% esto no tiene sentido, intencionalmente. Observe que las plicas siguen negras
\override Stem #'color = #(x11-color 'Boggle)
b2 cis
```



Se pueden especificar colores RGB exactos utilizando la función de Scheme `rgb-color`.

```
\override Staff.StaffSymbol #'color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}

\override Stem #'color = #(rgb-color 0 0 0)
gis8 a
\override Stem #'color = #(rgb-color 1 1 1)
gis8 a
\override Stem #'color = #(rgb-color 0 0 0.5)
gis4 a
```



Véase también

Referencia de la notación: Sección B.5 [Lista de colores], página 460, Sección 5.3.4 [La instrucción tweak], página 403.

Fragmentos de código: Sección "Editorial annotations" in *Fragmentos de código*.

Advertencias y problemas conocidos

Un color de X11 no es necesariamente de la misma tonalidad exacta que un color normal de nombre similar.
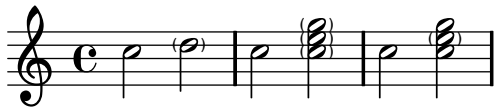
No todos los colores de X11 se distinguen entre sí en un navegador web, es decir, un navegador de web podría no mostrar ninguna diferencia entre `'LimeGreen` (verde lima) y `'ForestGreen` (verde bosque). Para la web se recomiendan los colores normales (o sea: `#blue`, azul, `#green`, verde, `#red`, rojo).

Las notas de un acorde no se pueden colorear con `\override`; en su lugar utilice `\tweak`. Consulte Sección 5.3.4 [La instrucción tweak], página 403 para ver más detalles.

Paréntesis
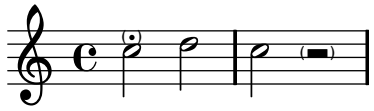
Los objetos se pueden encerrar entre paréntesis anteponiendo `\parenthesize` al evento musical. Si se aplica a un acorde, encierra cada nota dentro de un par de paréntesis. También se pueden poner entre paréntesis las notas individuales de un acorde.

```
c2 \parenthesize d
c2 \parenthesize <c e g>
c2 <c \parenthesize e g>
```



Los objetos que no son notas también se pueden poner entre paréntesis.

```
c2-\parenthesize -. d
c2 \parenthesize r
```



Véase también

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funciionamiento interno: Sección “Parenthesis_engraver” in *Referencia de Funcionamiento Interno*, Sección “ParenthesesItem” in *Referencia de Funcionamiento Interno*, Sección “parentheses-interface” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Al poner un acorde entre paréntesis, se encierra cada una de las notas individuales entre paréntesis, en vez de un solo paréntesis grande rodeando al acorde completo.

Plicas

Cuando se encuentra con una nota, se crea automáticamente un objeto `Stem` (plica). Para las redondas y los silencios, también se crean pero se hacen invisibles.

Instrucciones predefinidas
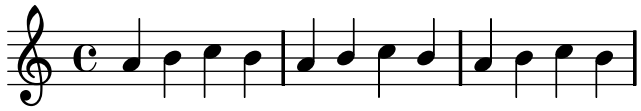
`\stemUp`, `\stemDown`, `\stemNeutral`.

Fragmentos de código seleccionados

Dirección predeterminada de las plicas sobre la tercera línea del pentagrama

La dirección predeterminada de las plicas sobre la tercera línea del pentagrama está determinada por la propiedad `neutral-direction` del objeto `Stem`.

```
\relative c'' {
  a4 b c b
  \override Stem #'neutral-direction = #up
  a4 b c b
  \override Stem #'neutral-direction = #down
  a4 b c b
}
```



Véase también

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 407.

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Stem_engraver” in *Referencia de Funcionamiento Interno*, Sección “Stem” in *Referencia de Funcionamiento Interno*, Sección “stem-interface” in *Referencia de Funcionamiento Interno*.
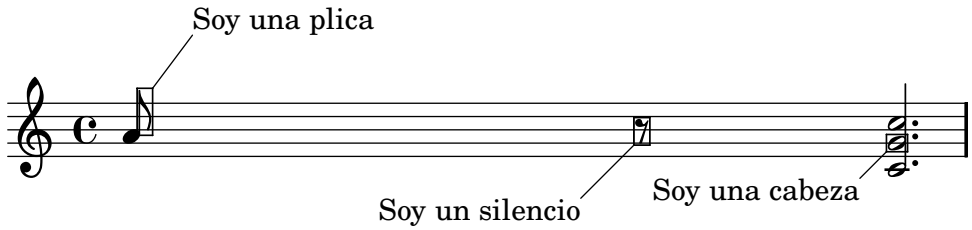
1.7.2 Fuera del pentagrama

Esta sección trata sobre cómo enfatizar elementos que están dentro del pentagrama, desde fuera del pentagrama.

Globos de ayuda

Los elementos de notación se pueden marcar y nombrar con la ayuda de un cartel o globo de ayuda rectangular. El propósito principal de esta funcionalidad es la explicación de la notación.

```
\new Voice \with { \consists "Balloon_engraver" }
{
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Soy una plica" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Soy un silencio" }
  r
  <c, g'-\balloonText #'(-2 . -2) \markup { "Soy una cabeza" } c>2.
}
```

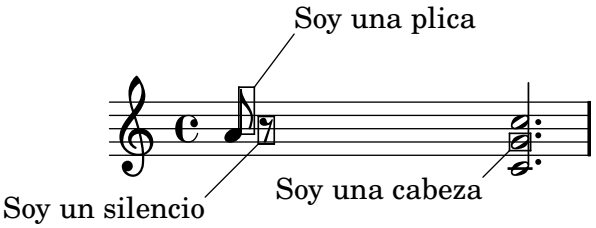


Existen dos funciones musicales, `balloonGrobText` y `balloonText`; el primero se usa como `\once\override` para adjuntar un texto a cualquier grob, y el último se usa coqmo `\tweak`, normalmente dentro de acordes, para adjuntar un texto a una nota individual.

Los textos de globo normalmente influyen en el espaciado de las notas, pero esto puede cambiarse:

```
\new Voice \with { \consists "Balloon_engraver" }
{
```

```
  \balloonLengthOff
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Soy una plica" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Soy un silencio" }
  r
  \balloonLengthOn
  <c, g'-\balloonText #'(-2 . -2) \markup { "Soy una cabeza" } c>2.
}
```



Instrucciones predefinidas

`\balloonLengthOn`, `\balloonLengthOff`.

Véase también

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Balloon_engraver” in *Referencia de Funcionamiento Interno*, Sección “BalloonTextItem” in *Referencia de Funcionamiento Interno*, Sección “balloon-interface” in *Referencia de Funcionamiento Interno*.

Líneas de rejilla

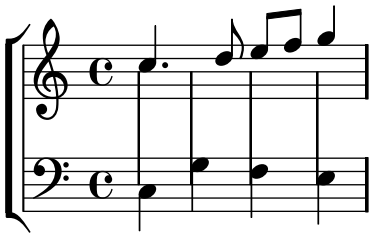
Se pueden dibujar líneas verticales entre los pentagramas sincronizadas con las notas.

Se debe usar el grabador `Grid_point_engraver` para crear los puntos extremos de las líneas, mientras que el grabador `Grid_line_span_engraver` se debe utilizar para trazar efectivamente las líneas. De forma predeterminada, esto centra las líneas de rejilla horizontalmente debajo y al lado izquierdo de la cabeza de las notas. Las líneas de rejilla se extienden a partir de línea media de los pentagramas. El intervalo `gridInterval` debe especificar la duración entre las líneas de rejilla.

```
\layout {
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #(ly:make-moment 1 4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
  }
}

\score {
  \new ChoirStaff <<
    \new Staff \relative c'' {
      \stemUp
      c4. d8 e8 f g4
```

```
    }
    \new Staff \relative c {
      \clef bass
      \stemDown
      c4 g' f e
    }
  >>
}
```



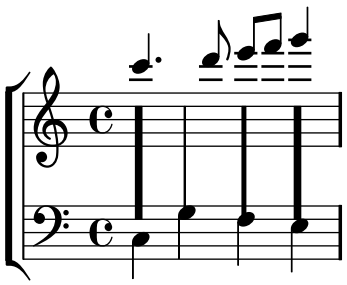
Fragmentos de código seleccionados

Líneas de rejilla: modificar su aspecto

Se puede cambiar el aspecto de las líneas de rejilla sobreescribiendo algunas de sus propiedades.

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \relative c'' {
        \stemUp
        c'4. d8 e8 f g4
      }
    }
    \new Staff {
      \relative c {
        % moverlas hacia arriba un espacio de pentagrama desde la posición de
        \override Score.GridLine #'extra-offset = #'(0.0 . 1.0)
        \stemDown
        \clef bass
        \once \override Score.GridLine #'thickness = #5.0
        c4
        \once \override Score.GridLine #'thickness = #1.0
        g'4
        \once \override Score.GridLine #'thickness = #3.0
        f4
        \once \override Score.GridLine #'thickness = #5.0
        e4
      }
    }
  >>
  \layout {
    \context {
      \Staff
      % establecer rejillas
      \consists "Grid_point_engraver"
```

```
      % establecer intervalo de rejilla a una negra
      gridInterval = #(ly:make-moment 1 4)
    }
    \context {
      \Score
      \consists "Grid_line_span_engraver"
      % moverlas a la derecha medio espacio de pentagrama
      \override NoteColumn #'X-offset = #-0.5
    }
  }
}
```



Véase también
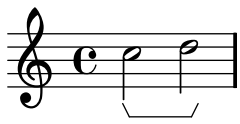
Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Grid_line_span_engraver” in *Referencia de Funcionamiento Interno*, Sección “Grid_point_engraver” in *Referencia de Funcionamiento Interno*, Sección “GridLine” in *Referencia de Funcionamiento Interno*, Sección “GridPoint” in *Referencia de Funcionamiento Interno*, Sección “grid-line-interface” in *Referencia de Funcionamiento Interno*, Sección “grid-point-interface” in *Referencia de Funcionamiento Interno*.

Corchetes de análisis

Los corchetes se usan en análisis musical para indicar la estructura de las piezas musicales. Están contemplados los corchetes horizontales simples.

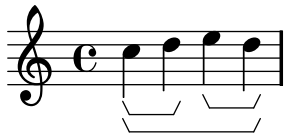
```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c2\startGroup
  d\stopGroup
}
```



Los corchetes de análisis se pueden anidar.

```
\layout {
  \context {
```

```
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c4\startGroup\startGroup
  d4\stopGroup
  e4\startGroup
  d4\stopGroup\stopGroup
}
```
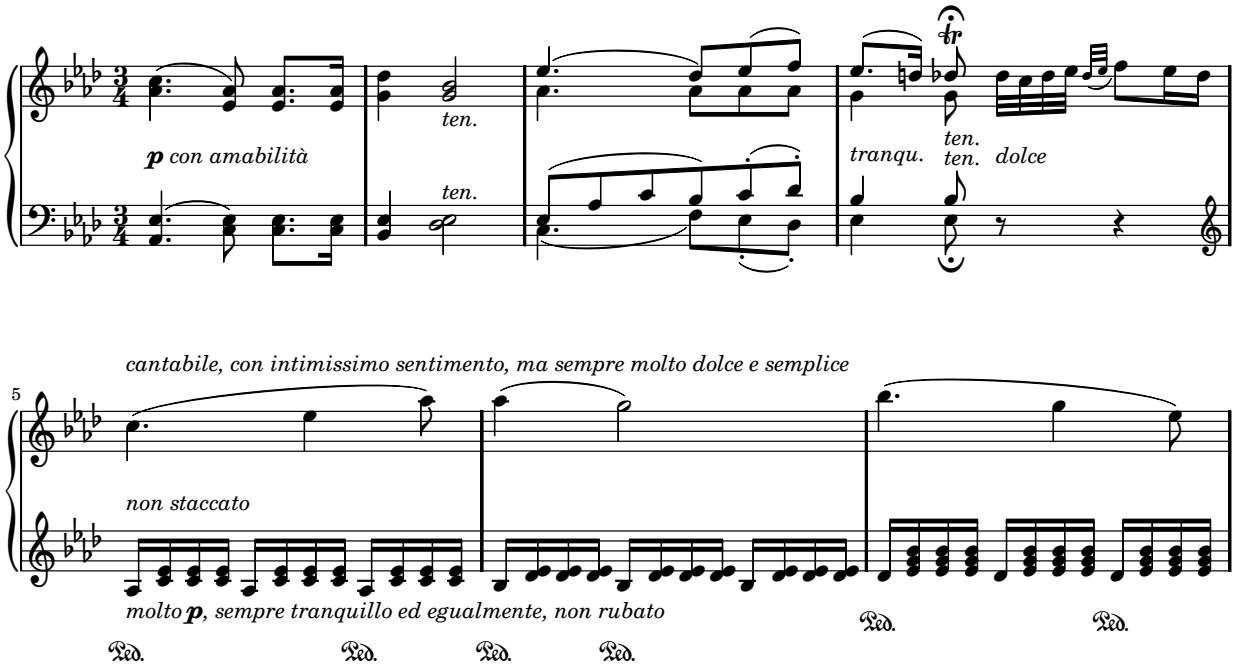


Véase también

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Horizontal_bracket_engraver” in *Referencia de Funcionamiento Interno*, Sección “HorizontalBracket” in *Referencia de Funcionamiento Interno*, Sección “horizontal-bracket-interface” in *Referencia de Funcionamiento Interno*, Sección “Staff” in *Referencia de Funcionamiento Interno*.

1.8 Texto



Esta sección explica cómo incluir texto (con diversos estilos de formateo) en las partituras.

Ciertos elementos de texto de los que no nos ocupamos aquí se tratan en otras secciones específicas: Sección 2.1 [Música vocal], página 190, Sección 3.2 [Títulos y cabeceras], página 320.

1.8.1 Escritura del texto
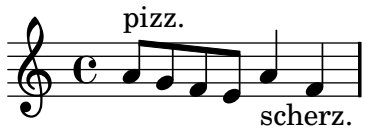
Esta sección presenta las distintas formas de añadir texto a una partitura.

Nota: Para escribir texto con caracteres acentuados y especiales (como los de idiomas distintos del inglés), sencillamente inserte los caracteres directamente en el archivo de LilyPond. El archivo se debe guardar como UTF-8. Para ver más información, consulte Sección 3.3.3 [Codificación del texto], página 332.

Guiones de texto
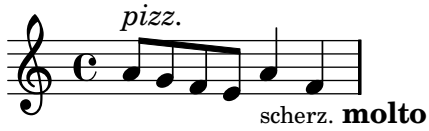
Es posible añadir indicaciones de "texto entrecomillado" a una partitura, como se muestra en el ejemplo siguiente. Tales indicaciones se pueden colocar también manualmente encima o debajo del pentagrama utilizando la sintaxis que se describe en Sección 5.4.2 [Dirección y posición], página 407.

```
a8^"pizz." g f e a4-"scherz." f
```
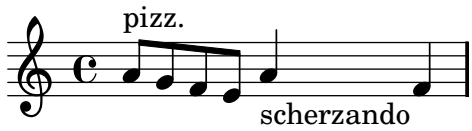


Esta sintaxis es en realidad una abreviatura; se puede añadir explícitamente a una nota un formateado de texto más complejo utilizando un bloque `\markup`, como se describe bajo Sección 1.8.2 [Formatear el texto], página 173.

```
a8^\markup { \italic pizz. } g f e
a4_\markup { \tiny scherz. \bold molto } f
```



De forma predeterminada, las indicaciones de texto no afectan al espaciado de las notas. Sin embargo, su anchura sí podría tenerse en cuenta: en el siguiente ejemplo, la primera cadena de texto no afecta al espaciado, pero la segunda sí afecta.

```
a8^"pizz." g f e
\textLengthOn
a4_"scherzando" f
```



Instrucciones predefinidas

`\textLengthOn`, `\textLengthOff`.

Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 173, Sección 5.4.2 [Dirección y posición], página 407.

Fragmentos de código: Sección "Text" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "TextScript" in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

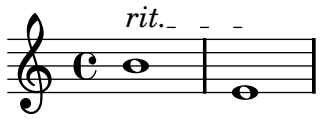
La comprobación necesaria para asegurar que las inscripciones de texto y la letra de las canciones se mantienen dentro de los márgenes, es una tarea computacional relativamente grande. Para acelerar el procesado, LilyPond no realiza estos cálculos de forma predeterminada; para habilitarlos, utilice

```
\override Score.PaperColumn #'keep-inside-line = ##t
```

Extensiones de texto

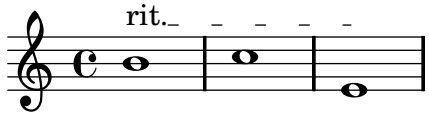
Ciertas indicaciones de ejecución, p.ej., *rallentando* o *accelerando*, se escriben como texto y se extienden sobre muchos compases mediante líneas de puntos. Estos objetos, que reciben el nombre de 'spanners' u objetos de extensión, se pueden crear dede una nota hasta otra usando la siguiente sintaxis:

```
\override TextSpanner #'(bound-details left text) = "rit."
b1\startTextSpan
e,\stopTextSpan
```



La cadena de texto que imprimir se establece a través de propiedades de objeto. De forma predeterminada se imprime en estilo cursiva, pero se pueden conseguir distintos efectos de formato utilizando bloques `\markup`, como se describe en Sección 1.8.2 [Formatear el texto], página 173.

```
\override TextSpanner #'(bound-details left text) =
  \markup { \upright "rit." }
b1\startTextSpan c
e,\stopTextSpan
```



El estilo de la línea, así como la cadena de texto, se pueden definir como una propiedad de objeto. Esta sintaxis se describe en Sección 5.4.7 [Estilos de línea], página 419.

Instrucciones predefinidas

`\textSpannerUp`, `\textSpannerDown`, `\textSpannerNeutral`.

Véase también

Referencia de la notación: Sección 5.4.7 [Estilos de línea], página 419, [Matices dinámicos], página 86.
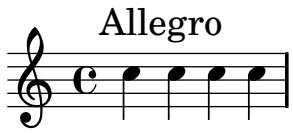
Fragmentos de código: Sección "Text" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "TextSpanner" in *Referencia de Funcionamiento Interno*.

Indicaciones de texto
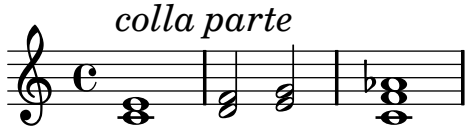
Se pueden añadir a la partitura varios elementos de texto con la sintaxis que se describe en [Llamadas de ensayo], página 77:

```
c4
\mark "Allegro"
c c c
```
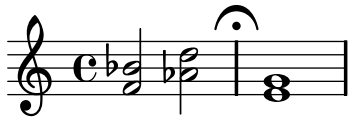


Esta sintaxis posibilita colocar cualquier texto sobre la línea divisoria; se pueden incorporar formatos más complejos para el texto usando un bloque `\markup`, como está explicado en Sección 1.8.2 [Formatear el texto], página 173:

```
<c e>1
\mark \markup { \italic { colla parte } }
<d f>2 <e g>
<c f aes>1
```
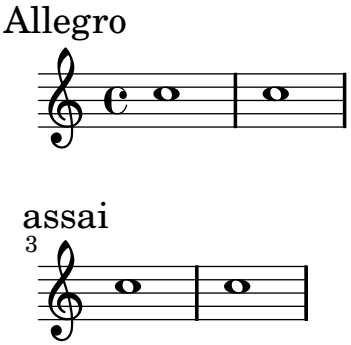


Esta sintaxis también permite imprimir símbolos especiales como llamadas, segno o calderones, especificando el nombre del símbolo correspondiente como se esplica en [Notación musical dentro de elementos de marcado], página 183:

```
<bes f>2 <aes d>
\mark \markup { \musicglyph #"scripts.ufermata" }
<e g>1
```



Estos objetos se tipografían solamente sobre el pentagrama superior de la partitura; dependiendo de si están especificados al final o en medio de un compás, se colocarán sobre la línea divisoria o entre las notas. Si se especifican en un salto de línea, las llamadas se imprimen al principio de la línea siguiente.

```
\mark "Allegro"
c1 c
\mark "assai" \break
c  c
```
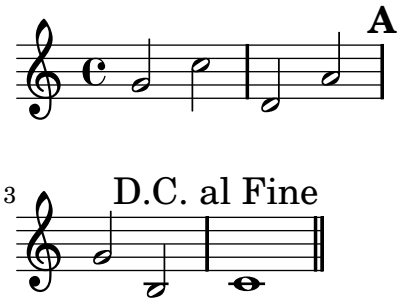


Fragmentos de código seleccionados

Imprimir marcas al final de la línea o de la partitura

Las marcas se pueden imprimir al final de la línea actual, en vez de al comienzo de la línea siguiente. Esto es útil principalmente cuando se tiene que añadir una marca al final de la partitura (donde no hay línea siguiente).

En tales casos, el extremo derecho de la marca se tiene que alinear con la última línea divisoria, como muestra la segunda línea de este ejemplo.

```
\relative c'' {
  \override Score.RehearsalMark #'break-visibility = #begin-of-line-invisible
  g2 c
  d,2 a'
  \mark \default
  \break
  g2 b,
  c1 \bar "||"
  \override Score.RehearsalMark #'self-alignment-X = #RIGHT
  \mark "D.C. al Fine"
}
```



Alinear marcas con varios objetos de notación

Si están especificadas, las marcas de texto se pueden alinear con objetos de notación distintos a las líneas divisorias. Entre estos objetos se encuentran `ambitus`, `breathing-sign`, `clef`, `custos`, `staff-bar`, `left-edge`, `key-cancellation`, `key-signature` y `time-signature`.

En estos casos, las marcas de texto se centran horizontalmente sobre el objeto, aunque esto se puede cambiar, como se muestra en la segunda línea de este ejemplo (en una partitura con varios pentagramas, se debe hacer este ajuste para todos los pentagramas).

```
\relative c' {
  e1

  % la RehearsalMark se centra sobre la clave
  \override Score.RehearsalMark #'break-align-symbols = #'(clef)
  \key a \major
  \clef treble
  \mark ""
  e1

  % la RehearsalMark se centra sobre el compás
  \override Score.RehearsalMark #'break-align-symbols = #'(time-signature)
  \key a \major
  \clef treble
  \time 3/4
```

```
  \mark ""
  e2.

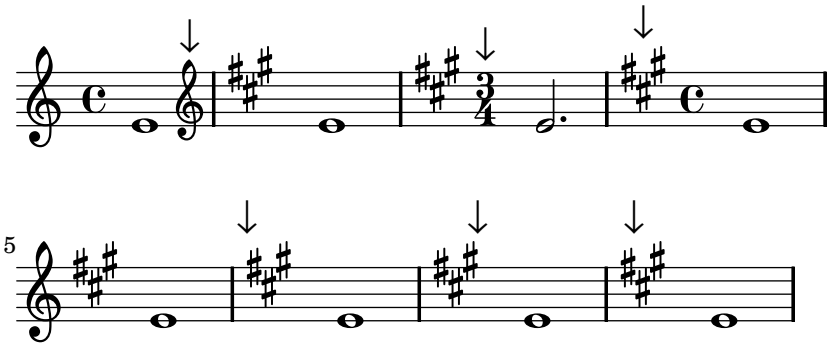
  % la RehearsalMark se centra sobre la armadura
  \override Score.RehearsalMark #'break-align-symbols = #'(key-signature)
  \key a \major
  \clef treble
  \time 4/4
  \mark ""
  e1

  \break
  e1

  % la RehearsalMark se alinea por la izquierda con la armadura
  \once \override Score.KeySignature #'break-align-anchor-alignment = #LEFT
  \mark ""
  \key a \major
  e1

  % la RehearsalMark se alinea por la derecha con la armadura
  \once \override Score.KeySignature #'break-align-anchor-alignment = #RIGHT
  \key a \major
  \mark ""
  e1

  % la RehearsalMark se alinea por la izquierda con la armadura
  % y se desplaza una unidad a la derecha.
  \once \override Score.KeySignature #'break-align-anchor = #1
  \key a \major
  \mark ""
  e1
}
```

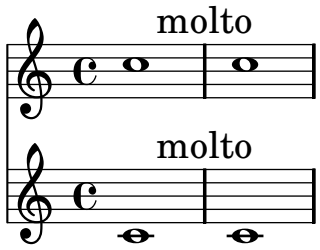


Imprimir marcas en cualquier pentagrama

Aunque normalmente las marcas de texto sólo se imprimen sobre el pentagrama superior, también se pueden imprimir en otro pentagrama cualquiera.

```
\score {
  <<
    \new Staff { c''1 \mark "molto" c'' }
    \new Staff { c'1 \mark "molto" c' }
```

```
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
    \context {
      \Staff
      \consists "Mark_engraver"
      \consists "Staff_collecting_engraver"
    }
  }
}
```



Véase también

Referencia de la notación: [Llamadas de ensayo], página 77, Sección 1.8.2 [Formatear el texto], página 173, [Notación musical dentro de elementos de marcado], página 183, Sección B.6 [La tipografía Feta], página 461.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RehearsalMark” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Si se introduce una llamada al final del último compás de la partitura (donde no hay línea siguiente), la llamada no se imprime en absoluto.

Texto separado

Un bloque `\markup` puede existir de forma independiente, fuera de cualquier bloque `\score`, como una “expresión de nivel superior”. Esta sintaxis se describe en Sección 3.1.3 [Estructura del archivo], página 318.

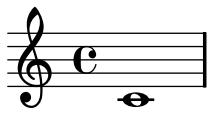
```
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
```

Tomorrow, and tomorrow, and tomorrow...

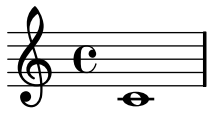
Esto hace posible imprimir texto separado de la música, lo que es especialmente útil cuando el archivo de entrada contiene varias piezas, tal y como se describe en Sección 3.1.2 [Varias partituras en un libro], página 317.

```
\score {
  c'1
```

```
}
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
\score {
  c'1
}
```



Tomorrow, and tomorrow, and tomorrow...



Los bloques de texto independientes pueden abarcar varias páginas, posibilitando la impresión de documentos de texto o libros íntegramente desde LilyPond. Esta funcionalidad y la sintaxis que requiere se describen en [Elementos de marcado de varias páginas], página 185.

Instrucciones predefinidas

`\markup`, `\markuplines`.

Fragmentos de código seleccionados

Elemento de marcado de texto independiente en dos columnas

Los textos independientes se pueden disponer en varias columnas utilizando instrucciones `\markup`:

```
\markup {
 \fill-line {
  \hspace #1.0
  \column {
   \line {"O sacrum convivium" }
   \line {"in quo Christus sumitur," }
   \line {"recolitur memoria passionis ejus," }
   \line {"mens impletur gratia," }
   \line {"futurae gloriae nobis pignus datur." }
   \line {"Amen."}
  }
  \hspace #2
  \column {
   \line { \italic {"O sacred feast"} }
   \line { \italic {"in which Christ is received,"} }
   \line { \italic {"the memory of His Passion is renewed,"} }
   \line { \italic {"the mind is filled with grace," } }
   \line { \italic {"and a pledge of future glory is given to us." }}
   \line { \italic {"Amen."}}
  }
  \hspace #1.0
```

```
 }
}
```

O sacrum convivium	*O sacred feast*
in quo Christus sumitur,	*in which Christ is received,*
recolitur memoria passionis ejus,	*the memory of His Passion is renewed,*
mens impletur gratia,	*the mind is filled with grace,*
futurae gloriae nobis pignus datur.	*and a pledge of future glory is given to us.*
Amen.	*Amen.*

Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 173, Sección 3.1.3 [Estructura del archivo], página 318, Sección 3.1.2 [Varias partituras en un libro], página 317, [Elementos de marcado de varias páginas], página 185.

Fragmentos de código: Sección "Text" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "TextScript" in *Referencia de Funcionamiento Interno*.

1.8.2 Formatear el texto

Esta sección presenta los formateados básico y avanzado de texto, usando la sintaxis específica del modo de `\markup`.

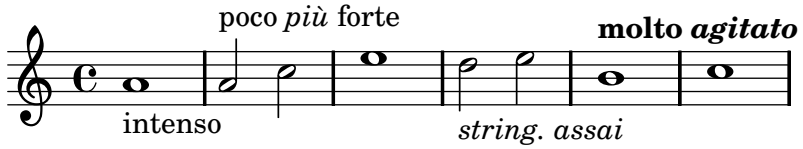
Introducción al marcado de texto

Se usa un bloque `\markup` para tipografiar texto con una sintaxis ampliable que se denomina "modo de marcado".

La sintaxis del marcado es similar a la sintaxis usual de LilyPond: una expresión `\markup` se encierra entre llaves `{ ... }`. Una sola palabra se considera como una expresión mínima, y por tanto no necesita estar encerrada entre llaves.
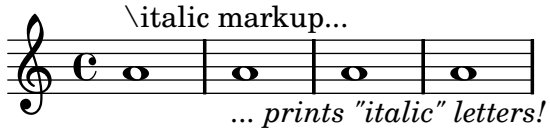
A diferencia de las indicaciones de "texto entrecomillado" simples, los bloques `\markup` pueden contener expresiones anidadas o instrucciones de marcado, que se introducen utilizando el carácter de barra invertida `\` . Estas instrucciones sólo afectan a la expresión que sigue inmediatamente.

```
a1-\markup intenso
a2^\markup { poco \italic più forte  }
c e1
d2_\markup { \italic "string. assai" }
e
b1^\markup { \bold { molto \italic  agitato } }
c
```



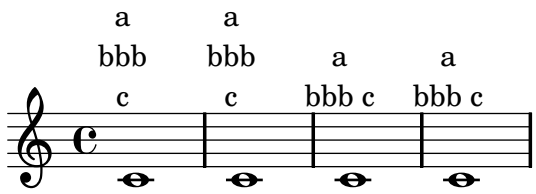
Un bloque `\markup` puede contener también cadenas de texto entre comillas. Dichas cadenas se tratan como expresiones de texto mínimas, y por tanto cualquier instrucción de marcado o carácter especial (como `\` y `#`) se imprimen literalmente sin afectar al formateo del texto. Se puede hacer que se impriman las propias comillas si se les antepone una barra invertida.

```
a1^"\italic markup..."
a_\markup { \italic "... prints \"italic\" letters!" }
a a
```



Para que se traten como una expresión distinta, es necesario que las listas de palabras estén encerradas entre comillas o precedidas de una instrucción. La forma en que están definidas las expresiones de marcado afecta a cómo se apilan, se centran y se alinean estas expresiones; en el ejemplo siguiente, la segunda expresión `\markup` se trata igual que la primera:
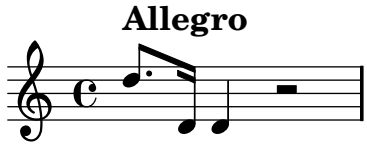
```
c1^\markup { \center-column { a bbb c } }
c1^\markup { \center-column { a { bbb c } } }
c1^\markup { \center-column { a \line { bbb c } } }
c1^\markup { \center-column { a "bbb c" } }
```



Los marcados se pueden almacenar dentro de variables. Estas variables se pueden adjuntar directamente a las notas:

```
allegro = \markup { \bold \large Allegro }

{
  d''8.^\allegro
  d'16 d'4 r2
}
```



Se puede encontrar una lista exhaustiva de las instrucciones específicas de `\markup` en Sección B.8 [Text markup commands], página 476.

Véase también

Referencia de la notación: Sección B.8 [Text markup commands], página 476.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Archivos de inicio: ‘`scm/markup.scm`’.
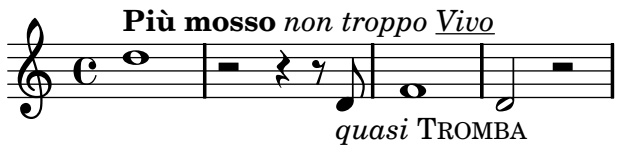
Advertencias y problemas conocidos

Los errores de sintaxis para el modo de marcado a menudo producen confusión.

Instrucciones de marcado de texto más usuales

Está contemplado de forma básica el cambio de la fuente tipográfica en el modo de marcado:

```
d1^\markup {
  \bold { Più mosso }
  \italic { non troppo \underline Vivo }
}
r2 r4 r8
d,_\markup { \italic quasi \smallCaps Tromba }
f1 d2 r
```
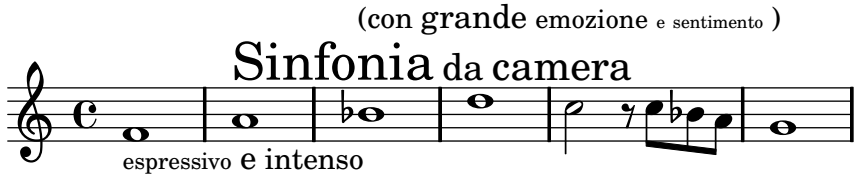


El tamaño de los caracteres también se puede alterar de distintas maneras:

- el tamaño de la tipografía se puede definir a una serie de tamaños estándar predefinidos,
- el tamaño de la tipografía se puede establecer a un valor absoluto,
- el tamaño de la tipografía también se puede cambiar de forma relativa respecto de su valor anterior.

El ejemplo siguiente ejemplifica estos tres métodos:

```
f1_\markup {
  \tiny espressivo
  \large e
  \normalsize intenso
}
a^\markup {
  \fontsize #5 Sinfonia
  \fontsize #2 da
  \fontsize #3 camera
}
bes^\markup { (con
  \larger grande
  \smaller emozione
  \magnify #0.6 { e sentimento } )
}
d c2 r8 c bes a g1
```



El texto se puede imprimir como subíndice o como superíndice. De forma predeterminada se imprimen en un tamaño menor, pero también se puede usar un tamaño normal:

```
\markup {
  \column {
    \line { 1 \super st movement }
    \line { 1 \normal-size-super st movement
```

```
      \sub { (part two) }  }
  }
}
```

1st movement
1st movement (part two)

El modo de marcado ofrece una manera fácil de elegir familias de tipografía alternativas. A no ser que se especifique de otro modo, se selecciona automáticamente la fuente predeterminada con serifa, de tipo romano: en la última línea del ejemplo siguiente, no hay diferencia entre la primera y la segunda palabra.

```
\markup {
  \column {
    \line { Act \number 1 }
    \line { \sans { Scene I. } }
    \line { \typewriter { Verona. An open place. } }
    \line { Enter \roman Valentine and Proteus. }
  }
}
```

Act 1
Scene I.
Verona. An open place.
Enter Valentine and Proteus.

Algunas de estas familias, usadas para elementos específicos como números o matices, no ofrecen todos los caracteres, como se explica en [Indicaciones dinámicas contemporáneas], página 90 y [Marcas de repetición manual], página 105.

Si se usan dentro de una palabra, algunas instrucciones de cambio de tipografía o de formateo pueden producir un espacio vacío no deseado. Esto se puede solucionar concatenando en uno solo los distintos elementos de texto:

```
\markup {
  \column {
    \line {
      \concat { 1 \super st }
      movement
    }
    \line {
      \concat { \dynamic p , }
      \italic { con dolce espressione }
    }
  }
}
```

1st movement
p, *con dolce espressione*

Se puede encontrar una lista exhaustiva de instrucciones de cambio y utilización personalizada de las fuentes tipográficas en Sección B.8.1 [Font], página 476.

También es posible definir conjuntos personalizados de fuentes, tal y como se explica en Sección 1.8.3 [Tipografías], página 186.

Instrucciones predefinidas

`\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`, `\smaller`, `\larger`.

Véase también

Referencia de la notación: Sección B.8.1 [Font], página 476, [Indicaciones dinámicas contemporáneas], página 90, [Marcas de repetición manual], página 105, Sección 1.8.3 [Tipografías], página 186.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

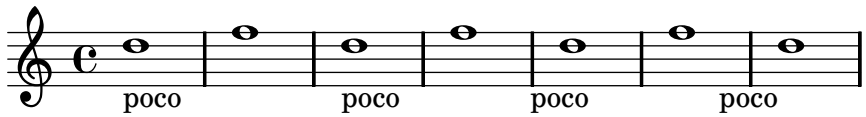
Archivos de inicio: ‘`scm/define-markup-commands.scm`’.

Alineación de texto

Esta subsección trata sobre cómo colocar texto en el modo de marcado. Los objetos de marcado también se pueden mover como un todo, usando la sintaxis que se describe en Sección “Moving objects” in *Manual de Aprendizaje*.
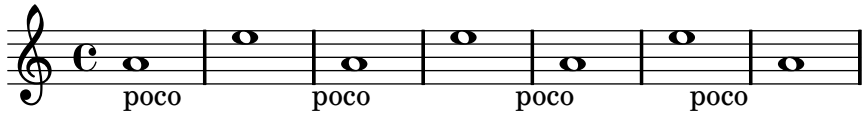
Los objetos de marcado se pueden alinear de distintas maneras. De foma predeterminada, una indicación de texto se alinea sobre el borde izquierdo: en el ejemplo siguiente, no existe diferencia entre los marcados primero y segundo.

```
d1-\markup { poco }
f
d-\markup { \left-align poco }
f
d-\markup { \center-align { poco } }
f
d-\markup { \right-align poco }
```



Se puede realizar un ajuste fino de la alineación horizontal usando un valor numérico:

```
a1-\markup { \halign #-1 poco }
e'
a,-\markup { \halign #0 poco }
e'
a,-\markup { \halign #0.5 poco }
e'
a,-\markup { \halign #2 poco }
```

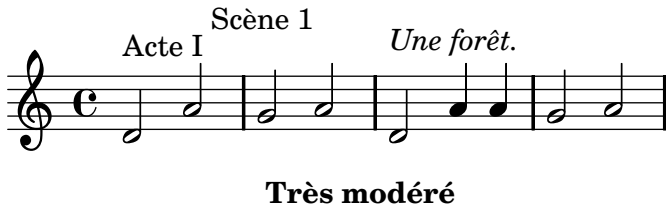


Ciertos objetos pueden poseer sus propios procedimientos de alineación, y por tanto no resultan afectados por estas instrucciones. Es posible mover estos objetos de marcado como un todo, como se muestra por ejemplo en [Indicaciones de texto], página 167.

La alineación vertical es un poco más compleja. Como se ha dicho más arriba, los objetos de marcado se pueden mover como un todo; sin embargo, también es posible mover elementos
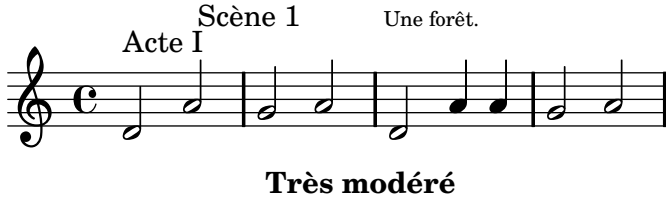
específicos dentro de un bloque de marcado. En este caso, el elemento a mover se debe preceder de un *punto de anclaje*, que puede ser otro elemento de marcado o un objeto invisible. El ejemplo siguiente ejemplifica estas dos posibilidades; el último marcado en este ejemplo no tiene punto de anclaje y por ello no resulta movido.

```
d2^\markup {
  Acte I
  \raise #2 { Scène 1 }
}
a'
g_\markup {
  \null
  \lower #4 \bold { Très modéré }
}
a
d,^\markup {
  \raise #4 \italic { Une forêt. }
}
a'4 a g2 a
```



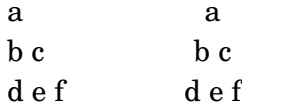
Algunas instrucciones pueden afcetar tanto a la alineación horizontal como a la vertical de los objetos de texto en el modo de marcado. Cualquier objeto afectado por estas instrucciones se deben preceder de un punto de anclaje:

```
d2^\markup {
  Acte I
  \translate #'(-1 . 2) "Scène 1"
}
a'
g_\markup {
  \null
  \general-align #Y #3.2 \bold "Très modéré"
}
a
d,^\markup {
  \null
  \translate-scaled #'(-1 . 2) \teeny "Une forêt."
}
a'4 a g2 a
```



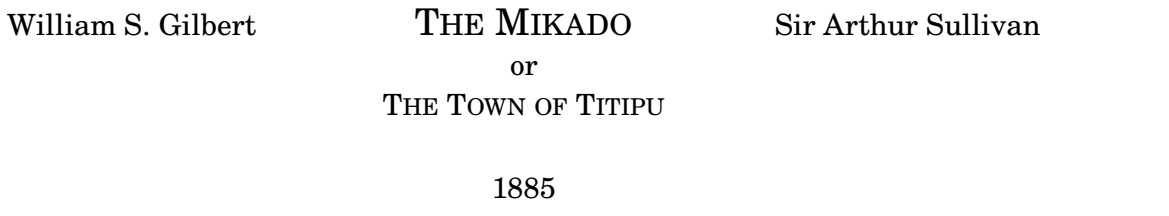
Un objeto de marcado puede incluir varias líneas de texto. En el ejemplo siguiente, cada elemento o expresión se sitúa en su propia línea, ya sea alineada por la izquierda o centrada:

```
\markup {
  \column {
    a
    "b c"
    \line { d e f }
  }
  \hspace #10
  \center-column {
    a
    "b c"
    \line { d e f }
  }
}
```



De forma similar, una lista de elementos o expresiones se puede repartir de forma que rellene por completo el ancho de la línea horizontal (si hay un solo elemento, se centra en el papel). A su vez, estas expresiones pueden incluir texto de varias líneas o cualquier otra expresión de marcado:

```
\markup {
  \fill-line {
    \line { William S. Gilbert }
    \center-column {
      \huge \smallCaps "The Mikado"
      or
      \smallCaps "The Town of Titipu"
    }
    \line { Sir Arthur Sullivan }
  }
}
\markup {
  \fill-line { 1885 }
}
```



Las indicaciones de texto largas se pueden también ajustar automáticamente según un ancho de línea dado. Estarán alineados por la izquierda o justificados, como se muestra en el ejemplo siguiente.

```
\markup {
  \column {
    \line  \smallCaps { La vida breve }
```

```
    \line \bold { Acto I }
    \wordwrap \italic {
      (La escena representa el corral de una casa de
      gitanos en el Albaicín de Granada. Al fondo una
      puerta por la que se ve el negro interior de
      una Fragua, iluminado por los rojos resplandores
      del fuego.)
    }
    \hspace #0

    \line \bold { Acto II }
    \override #'(line-width . 50)
    \justify \italic {
      (Calle de Granada. Fachada de la casa de Carmela
      y su hermano Manuel con grandes ventanas abiertas
      a través de las que se ve el patio
      donde se celebra una alegre fiesta)
    }
  }
}
```

La vida breve

Acto I

(La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.)

Acto II

(Calle de Granada. Fachada de la casa de Carmela y su hermano Manuel con grandes ventanas abiertas a través de las que se ve el patio donde se celebra una alegre fiesta)

Hay una lista exhaustiva de instrucciones de alineación de texto en Sección B.8.2 [Align], página 485.

Véase también

Manual de aprendizaje: Sección “Moving objects” in *Manual de Aprendizaje*.

Referencia de la notación: Sección B.8.2 [Align], página 485, [Indicaciones de texto], página 167.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

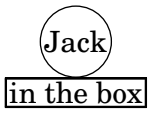
Archivos de inicio: ‘`scm/define-markup-commands.scm`’.

Notación gráfica dentro de elementos de marcado

Se puede añadir diversos objetos gráficos a una partitura, utilizando instrucciones de marcado.

Ciertas instrucciones de marcado permiten la decoración de elementos de texto con gráficos, como se muestra en el ejemplo siguiente.

```
\markup \fill-line {
  \center-column {
    \circle Jack
    \box "in the box"
    \null
    \line {
      Erik Satie
      \hspace #3
      \bracket "1866 - 1925"
    }
    \null
    \rounded-box \bold Prelude
  }
}
```



Erik Satie [1866 - 1925]

Prelude

Algunas instrucciones pueden requerir un aumento del relleno alrededor del texto: esto se puede conseguir con algunas instrucciones de marcado descritas exhaustivamente en Sección B.8.2 [Align], página 485.

```
\markup \fill-line {
  \center-column {
    \box "Charles Ives (1874 - 1954)"
    \null
    \box \pad-markup #2 "THE UNANSWERED QUESTION"
    \box \pad-x #8 "A Cosmic Landscape"
    \null
  }
}
\markup \column {
  \line {
    \hspace #10
    \box \pad-to-box #'(-5 . 20) #'(0 . 5)
      \bold "Largo to Presto"
  }
  \pad-around #3
      "String quartet keeps very even time,
Flute quartet keeps very uneven time."
}
```

Charles Ives (1874 - 1954)

THE UNANSWERED QUESTION

A Cosmic Landscape

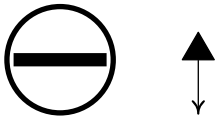
Largo to Presto

String quartet keeps very even time, Flute quartet keeps very uneven time.

Se pueden imprimir otros símbolos o elementos gráficos sin que se requiera ningún otro texto. De igual manera que en el caso de las expresiones de marcado, dichos objetos se pueden combinar:

```
\markup {
  \combine
    \draw-circle #4 #0.4 ##f
    \filled-box #'(-4 . 4) #'(-0.5 . 0.5) #1
  \hspace #5

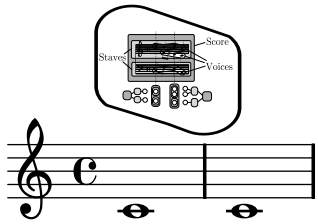
  \center-column {
    \triangle ##t
    \combine
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
  }
}
```



Entre las funcionalidades gráficas avanzadas se encuentran la posibilidad de incluir archivos de imagen externos convertidos al formato de PostScript Encapsulado (*eps*), y la inclusión directa de gráficos dentro del archivo de entrada, usando código PostScript nativo. En tal caso puede ser de utilidad la especificación explícita del tamaño del dibujo, como se ejemplifica a continuación:

```
c1^\markup {
  \combine
    \epsfile #X #10 #"./context-example.eps"
    \with-dimensions #'(0 . 6) #'(0 . 10)
    \postscript #"
      -2 3 translate
      2.7 2 scale
      newpath
      2 -1 moveto
      4 -2 4 1 1 arct
      4 2 3 3 1 arct
      0 4 0 3 1 arct
      0 0 1 -1 1 arct
      closepath
      stroke"
  }
```

```
c
```



Hay una lista exhaustiva de instrucciones específicas de gráficos en Sección B.8.3 [Graphic], página 498.

Véase también

Referencia de la notación: Sección B.8.3 [Graphic], página 498, [Anotaciones editoriales], página 309.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

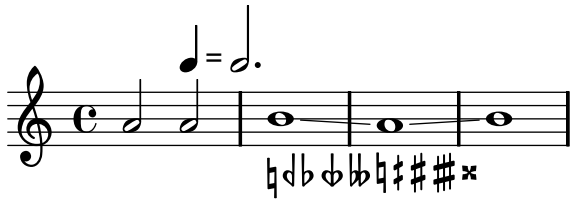
Archivos de inicio: ‘`scm/define-markup-commands.scm`’, ‘`scm/stencil.scm`’.

Notación musical dentro de elementos de marcado

Se pueden añadir a la partitura diversos elementos de notación musical, dentro de un objeto de marcado.

Las notas y las alteraciones se pueden escribir utilizando instrucciones de marcado:

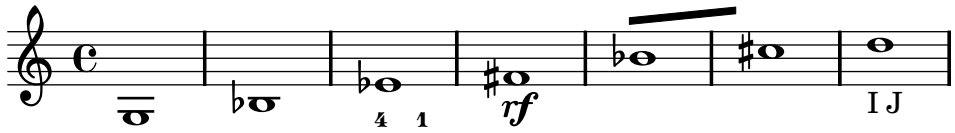
```
a2 a^\markup {
  \note #"4" #1
  =
  \note-by-number #1 #1 #1.5
}
b1_\markup {
  \natural \semiflat \flat
  \sesquiflat \doubleflat
}
\glissando
a1_\markup {
  \natural \semisharp \sharp
  \sesquisharp \doublesharp
}
\glissando b
```



Otros objetos de notación se pueden también imprimir en el modo de marcado:

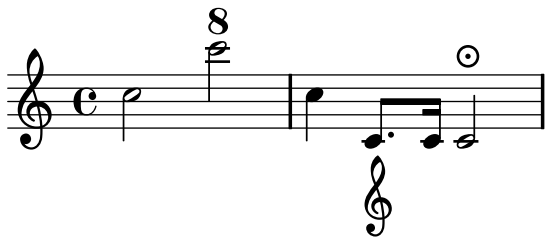
```
g1 bes
ees-\markup {
```

```
  \finger 4
  \tied-lyric #"~"
  \finger 1
}
fis_\markup { \dynamic rf }
bes^\markup {
  \beam #8 #0.1 #0.5
}
cis
d-\markup {
  \markalphabet #8
  \markletter #8
}
```



De forma más general, cualquier símbolo musical disponible se puede incluir por separado dentro de un objeto de marcado, como se ejemplifica a continuación; hay una lista exhaustiva de estos símbolos y sus nombres en Sección B.6 [La tipografía Feta], página 461.

```
c2
c'^\markup { \musicglyph #"eight" }
c,4
c,8._\markup { \musicglyph #"clefs.G_change" }
c16
c2^\markup { \musicglyph #"timesig.neomensural94" }
```

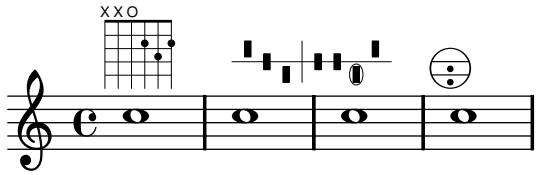


Otra forma de imprimir glifos que no son de texto se encuentra descrita en [Explicación de las fuentes tipográficas], página 186.

El modo de marcado también contempla diagramas para instrumentos específicos:

```
c1^\markup {
  \fret-diagram-terse #"x;x;o;2;3;2;"
}
c^\markup {
  \harp-pedal #"^-v|--ov^"
}
c
c^\markup {
  \combine
    \musicglyph #"accordion.accDiscant"
    \combine
      \raise #0.5 \musicglyph #"accordion.accDot"
```
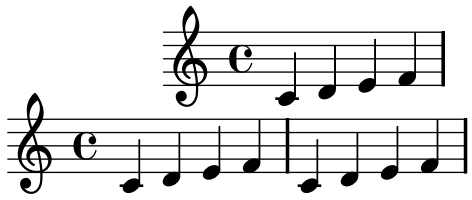
```
      \raise #1.5 \musicglyph #"accordion.accDot"
}
```



Dichos diagramas se encuentran documentados es Sección B.8.5 [Instrument Specific Markup], página 505.

Incluso una partitura completa se puede incluir dentro de un objeto de marcado. En tal caso, el bloque anidado `\score` debe contener un bloque `\layout`, como se muestra aquí:

```
c4 d^\markup {
  \score {
    \relative c' { c4 d e f }
    \layout { }
  }
}
e f |
c d e f
```



Hay una lista exhaustiva de instrucciones relacionadas con la notación musical en Sección B.8.4 [Music], página 502.

Véase también

Referencia de la notación: Sección B.8.4 [Music], página 502, Sección B.6 [La tipografía Feta], página 461, [Explicación de las fuentes tipográficas], página 186.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Archivos de inicio: ‘`scm/define-markup-commands.scm`’, ‘`scm/fret-diagrams.scm`’, ‘`scm/harp-pedals.scm`’.

Elementos de marcado de varias páginas

Aunque los objetos de marcado estándar no se pueden dividir, una sintaxis específica hace posible la introducción de líneas de texto que pueden abarcar varias páginas:

```
\markuplines {
  \justified-lines {
    Un texto muy largo de líneas justificadas.
    ...
  }
  \wordwrap-lines {
    Otro párrafo muy largo.
```

```
    ...
  }
  ...
}
```

Un texto muy largo de líneas justificadas. ...

Otro párrafo muy largo. ...

...

Esta sintaxis acepta una lista de elementos de marcado, que pueden ser

- el resultado de una instrucción de lista de marcado,
- una lista de marcados,
- o una lista de listas de marcado.

Hay una lista exhaustiva de las instrucciones de lista de marcado en Sección B.9 [Text markup list commands], página 512.

Véase también

Referencia de la notación: Sección B.9 [Text markup list commands], página 512, Sección 6.4.4 [Definición de nuevas instrucciones de lista de marcado], página 446.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” in *Referencia de Funcionamiento Interno*.

Archivos de inicio: ‘`scm/define-markup-commands.scm`’.

Instrucciones predefinidas

`\markuplines`.

1.8.3 Tipografías

Esta sección presenta la forma en que se manejan las fuentes tipográficas, y cómo se pueden cambiar en las partituras.
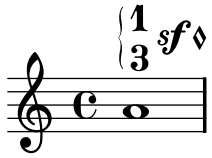
Explicación de las fuentes tipográficas

Las fuentes tipográficas se manejan a través de distintas bibliotecas. FontConfig se utiliza para detectar las tipografías disponibles en el sistema; las tipografías seleccionadas se trazan utilizando Pango.

Las tipografías de notación musical se pueden describir como un conjunto de glifos específicos, ordenados en varias familias. La siguiente sintaxis permite usar directamente varias tipografías `feta` de LilyPond distintas a las de texto en el modo de marcado:

```
a1^\markup {
  \vcenter {
    \override #'(font-encoding . fetaBraces)
    \lookup #"brace120"
    \override #'(font-encoding . fetaNumber)
    \column { 1 3 }
    \override #'(font-encoding . fetaDynamic)
    sf
```

```
    \override #'(font-encoding . fetaMusic)
    \lookup #"noteheads.s0petrucci"
  }
}
```
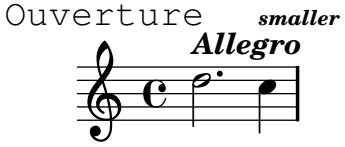


Una sintaxis más simple pero más limitada se describe en [Notación musical dentro de elementos de marcado], página 183.

Están disponibles tres familias de tipografías de texto: la tipografía *romana* (con serifa), que de forma predeterminada es New Century Schoolbook, la fuente *sans* y la fuente monoespaciada *de máquina de escribir*. Estas dos últimas familias vienen determinadas por la instalación de Pango.

Cada familia puede incluir distintas formas y series. El ejemplo siguiente muestra la posibilidad de seleccionar familias, formas, series y tamaños alternativos. El valor que se da a `font-size` es el cambio requerido a partir del tamaño predeterminado.

```
\override Score.RehearsalMark #'font-family = #'typewriter
\mark \markup "Ouverture"
\override Voice.TextScript #'font-shape = #'italic
\override Voice.TextScript #'font-series = #'bold
d2.^\markup "Allegro"
\override Voice.TextScript #'font-size = #-3
c4^smaller
```



Se puede usar una sintaxis similar en el modo de marcado, aunque en este caso es preferible usar la sintaxis sencilla que se explica en [Instrucciones de marcado de texto más usuales], página 175:

```
\markup {
  \column {
    \line {
      \override #'(font-shape . italic)
      \override #'(font-size . 4)
      Idomeneo,
    }
    \line {
      \override #'(font-family . typewriter)
      {
        \override #'(font-series . bold)
        re
        di
      }
      \override #'(font-family . sans)
      Creta
    }
```

```
  }
}
```

Idomeneo,
re di Creta

Aunque es fácil cambiar entre las fuentes tipográficas preconfiguradas, también es posible usar otras fuentes, como se explica en las siguientes secciones: [Fuentes de un solo elemento], página 188 y [Fuentes tipográficas del documento completo], página 188.

Véase también
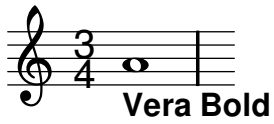
Referencia de la notación: Sección B.6 [La tipografía Feta], página 461, [Notación musical dentro de elementos de marcado], página 183, [Instrucciones de marcado de texto más usuales], página 175, Sección B.8.1 [Font], página 476.

Fuentes de un solo elemento

Se puede usar en una partitura cualquier fuente tipográfica que esté instalada en el sistema operativo y reconocida por parte de FontConfig, usando la siguiente sintaxis:

```
\override Staff.TimeSignature #'font-name = #"Charter"
\override Staff.TimeSignature #'font-size = #2
\time 3/4

a1_\markup {
  \override #'(font-name . "Vera Bold")
    { Vera Bold }
}
```



La instrucción siguiente presenta una lista de todas las tipografías disponibles en el sistema operativo:

```
lilypond -dshow-available-fonts x
```

El último argumento de la instrucción puede ser cualquiera, pero debe estar presente.

Véase también

Referencia de la notación: [Explicación de las fuentes tipográficas], página 186, [Fuentes tipográficas del documento completo], página 188.

Fragmentos de código: Sección “Text” in *Fragmentos de código*.

Archivos de inicio: ‘`lily/font-config-scheme.cc`’.

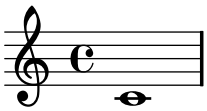
Fuentes tipográficas del documento completo

Se pueden cambiar las fuentes tipográficas que se usan como tipografías predeterminadas en las familias *romana*, *sans-serif* y *monoespaciada*, especificándolas en este orden como se ve en el ejemplo de abajo. Para leer más explicaciones sobre las fuentes tipográficas, consulte [Explicación de las fuentes tipográficas], página 186.

```
\paper  {
  myStaffSize = #20
  #(define fonts
```

```
    (make-pango-font-tree "Times New Roman"
                          "Nimbus Sans"
                          "Luxi Mono"
                           (/ myStaffSize 20)))
}

\relative c'{
  c1-\markup {
    roman,
    \sans sans,
    \typewriter typewriter. }
}
```



roman, sans, `typewriter.`

Véase también

Referencia de la notación: [Explicación de las fuentes tipográficas], página 186, [Fuentes de un solo elemento], página 188, [Instrucciones de marcado de texto más usuales], página 175, Sección B.8.1 [Font], página 476.

2 Notación especializada

Este capítulo explica cómo crear notación musical para instrumentos de un tipo específico o dentro de estilos específicos.

2.1 Música vocal

Esta sección explica cómo tipografiar música vocal, y cómo asegurarse de que la letra se alinea con las notas de su melodía correspondiente.

2.1.1 Notación común para música vocal

Esta sección trata sobre asuntos relacinados con la música vocal en general y con ciertos estilos concretos de música vocal.

Referencias para música vocal y letra

Pueden surgir varias cuestiones al tipografiar música vocal. Algunas se discuten en esta sección, pero otras están tratadas en otros lugares del manual:

- Casi todos los estilos de música vocal utilizan texto escrito como letra. Hay una introducción a esta notación en Sección “Setting simple songs” in *Manual de Aprendizaje*.
- La música vocal probablemente requiere el uso del modo de marcado o `markup`, ya sea para la letra o para otros elementos de texto (nombres de los personajes, etc.). Esta sintaxis está descrita en [Introducción al marcado de texto], página 173.
- Las hojas guía de acordes o *Lead sheets* se pueden imprimir combinando partes vocales y el ‘modo de acordes’; esta sintaxis se explica en Sección 2.7 [Notación de acordes], página 264.
- Los ‘ambitus’ o indicaciones de tesitura vocal se pueden añadir al principio de los pentagramas vocales, como se explica en [Tesitura], página 25.
- Las partes vocales se pueden imprimir utilizando las claves tradicionales, como se muestra en [Clave], página 12.
- Está contemplada la música vocal en estilo de notación antiguo, como se explica en Sección 2.8 [Notación antigua], página 283.

Ópera

Continuará...

Cancioneros

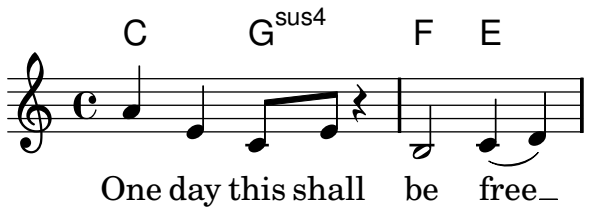
Continuaará...

Fragmentos de código seleccionados

Hoja guía de acordes o «lead sheet» sencilla

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet»:

```
<<
  \chords { c2 g:sus4 f e }
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Véase también

Referencia de la notación: Sección 2.7 [Notación de acordes], página 264.

Música hablada

Los efectos como el 'parlato' o el 'Sprechgesang' requieren de los intérpretes que hablen sin altura determinada pero con su ritmo; su notación se realiza mediante cabezas de nota en forma de aspas, como se muestra en [Cabezas de nota especiales], página 27.

Cantos

Continuará...

Música vocal antigua

Continuará...

Véase también

Referencia de la notación: Sección 2.8 [Notación antigua], página 283.

2.1.2 Introducir la letra

Explicación de la letra

Dado que los archivos de entrada de LilyPond son de texto, existe al menos un asunto que tener en cuenta al trabajar con música vocal: los textos de las canciones se deben interpretar como texto, no como notas. Por ejemplo, la entrada `d` debe interpretarse como una sílaba de una sola letra, no como la nota Re. Por tanto se hace necesario utilizar un modo especial para la letra, ya sea explícitamente o mediante el uso de métodos abreviados.

La letra de las canciones se introduce en un modo de entrada especial que se inicia mediante la palabra clave `\lyricmode`, o bien mediante `\addlyrics` ó `\lyricsto`. En este modo puede introducir la letra, con puntuación y acentos, y el carácter `d` no se analiza como una nota, sino más bien como una sílaba de una sola letra. Las sílabas se introducen como las notas, pero con texto en lugar de alturas de nota. Por ejemplo,

```
\lyricmode { Cam-4 pa-4 ni- ta del- lu- gar2 }
```

Existen dos métodos principales para especificar la colocación horizontal de las sílabas, bien especificando la duración de cada sílaba de forma explícita, como en el ejemplo anterior, o bien alineando automáticamente la letra a la melodía o a cualquier otra voz, usando `\addlyrics` o `\lyricsto`.

Una palabra o sílaba de la letra comienza con un carácter alfabético, y termina con cualquier espacio o dígito. Los caracteres que siguen pueden ser cualesquiera excepto un dígito o un espacio.

Cualquier carácter excepto un dígito o un espacio en blanco se considera parte de una sílaba; esto tiene una importante consecuencia, y es que una palabra puede terminar en `}`, lo que con frecuencia conduce al siguiente error:

```
\lyricmode { la- la}
```

En este ejemplo, el símbolo `}` se encuentra incluido dentro de la sílaba final, por tanto la llave de apertura no se compensa con la correspondiente llave de cierre y el archivo de entrada probablemente no se podrá procesar.

De forma similar, un punto que sigue a una secuencia alfabética queda incluido dentro de la cadena resultante. Como consecuencia, se deben insertar espacios antes y después de los comandos de propiedades: *no* escriba

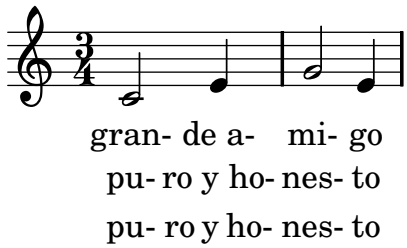
```
\override Score.LyricText #'font-shape = #'italic
```

sino lo siguiente:

```
\override Score . LyricText #'font-shape = #'italic
```

Para asignar más de una sílaba a una única nota, puede rodearlas por comillas o usar un carácter _ (guión bajo), para obtener espacios entre las sílabas, o usar el símbolo de tilde curva (~) para obtener una ligadura entre sílabas de la letra.
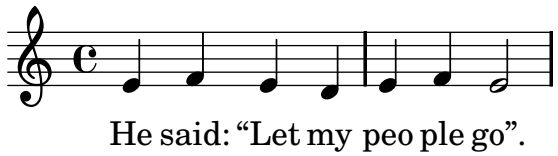
```
\time 3/4
\relative c' { c2 e4 g2 e4 }
\addlyrics { gran- de_a- mi- go }
\addlyrics { pu- "ro y ho-" nes- to }
\addlyrics { pu- ro~y~ho- nes- to }
```



La ligadura de texto se construye con el carácter Unicode U+203F, por tanto debe asegurarse de que tiene instalada una tipografía (como DejaVuLGC) que incluya este glifo.
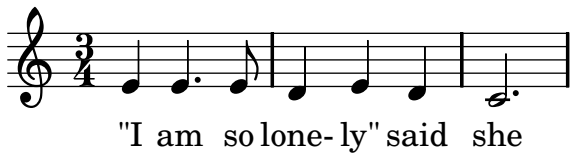
Para escribir letra con caracteres de una lengua no inglesa, o que tenga caracteres acentuados o especiales (como el símbolo del corazón o comillas inclinadas), introduzca simplemente los caracteres directamente en el archivo de entrada y guárdelo con una codificación utf-8. Consulte Sección 3.3.3 [Codificación del texto], página 332, para ver más información.

```
\relative c' { e4 f e d e f e2 }
\addlyrics { He said: \Let my peo ple go". }
```



Para utilizar comillas normales en la letra, escriba una barra invertida antes de las comillas. Por ejemplo,

```
\relative c' { \time 3/4 e4 e4. e8 d4 e d c2. }
\addlyrics { "\"I" am so lone- "ly\"" said she }
```



La definición completa del comienzo de una palabra en el modo Lyrics (letra) es algo más compleja.

Una palabra en el modo Lyrics comienza por: un carácter alfabético, `_`, `?`, `!`, `:`, `'`, los caracteres de control desde `^A` hasta `^F`, desde `^Q` hasta `^W`, `^Y`, `^^`, cualquier carácter de 8 bits con código ASCII por encima del 127, o una combinación de dos caracteres consistente en la combinación de una barra invertida seguida por `` ` ``, `'`, `"` ó `^`.

Para definir identificadores que contengan letra, se debe usar la función `lyricmode`.

```
verseOne = \lyricmode { Joy to the world the Lord is come }
\score {
  <<
    \new Voice = "one" \relative c'' {
      \autoBeamOff
      \time 2/4
      c4 b8. a16 g4. f8 e4 d c2
    }
    \addlyrics { \verseOne }
  >>
}
```

Véase también

Referencia de la notación: Sección 1.8.3 [Tipografías], página 186.

Referencia de funcionamiento interno: Sección “LyricText” in *Referencia de Funcionamiento Interno*, Sección “LyricSpace” in *Referencia de Funcionamiento Interno*.
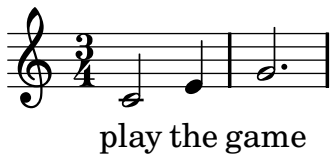
Elaborar canciones sencillas

La manera más fácil de poner letra a una melodía es añadir

```
\addlyrics { la letra }
```

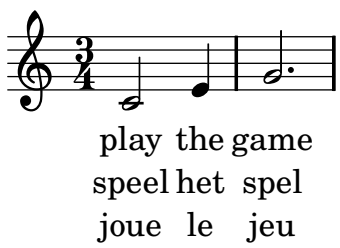
a la melodía. He aquí un ejemplo:

```
\time 3/4
\relative c' { c2 e4 g2. }
\addlyrics { play the game }
```



Se pueden añadir más versos mediante la adición de más secciones `\addlyrics`

```
\time 3/4
\relative c' { c2 e4 g2. }
\addlyrics { play the game }
\addlyrics { speel het spel }
\addlyrics { joue le jeu }
```



La instrucción `\addlyrics` no puede manejar situaciones de polifonía. Para estos casos tendrá que usar `\lyricsto` y `\lyricmode`, tal y como se explicará en [Explicación de la letra], página 191.

Trabajar con el texto y las variables

Para definir identificadores que contengan texto, se debe usar la función `\lyricmode`. Si embargo no tendrá que escribir las duraciones, si añade `\addlyrics` o `\lyricsto` en el momento de invocar el identificador.

```
verseOne = \lyricmode { Joy to the world the Lord is come }
\score {
 <<
   \new Voice = "one" \relative c'' {
     \autoBeamOff
     \time 2/4
     c4 b8. a16 g4. f8 e4 d c2
   }
   \addlyrics { \verseOne }
 >>
}
```

Para arreglos distintos o más complejos, la mejor forma es preparar previamente la jerarquía de pautas y textos, p.ej.:

```
\new ChoirStaff <<
  \new Voice = "soprano" { música }
  \new Lyrics = "textoSoprano" { s1 }
  \new Lyrics = "textoTenor" { s1 }
  \new Voice = "tenor" { música }
>>
```

y después combinar las melodías y líneas de texto correspondientes:

```
\context Lyrics = textoSoprano \lyricsto "soprano"
el texto
```

El código de entrada definitivo sería algo como

```
<<\new ChoirStaff << prepare la música >>
 \lyricsto "soprano" etc
 \lyricsto "alto" etc
etc
>>
```

Véase también

Referencia de funcionamiento interno: Sección “LyricCombineMusic” in *Referencia de Funcionamiento Interno*, Sección “Lyrics” in *Referencia de Funcionamiento Interno*.

2.1.3 Alineación de la letra a una melodía

La alineación del texto con las melodías se puede hacer automáticamente, pero si especificamos las duraciones de las sílabas también puede hacerse de forma manual. La alineación y el tipografiado de la letra se preparan con la ayuda de saltos o *skips*, guiones y líneas extensoras.

La letra se imprime a través de su interpretación dentro del contexto llamado Sección “Lyrics” in *Referencia de Funcionamiento Interno*.

```
\new Lyrics \lyricmode ...
```

Existen dos métodos principales para especificar la colocación horizontal de las sílabas:

- alineando automáticamente el texto a una melodía o a otra voz, utilizando `\addlyrics` o `\lyricsto`.
- o especificando la duración de cada sílaba explícitamente, utilizando `\lyricmode`

Duración automática de las sílabas

La letra se puede alinear automáticamente bajo una melodía dada. Esto se consigue combinando la melodía y el texto con la expresión `\lyricsto`

`\new Lyrics \lyricsto` *nombre* `...`

Esto alinea la letra a las notas del contexto de Sección "Voice" in *Referencia de Funcionamiento Interno* llamado *nombre*, que debe existir previamente. Por ello la `Voice` normalmente se especifica en primer lugar, y después se especifica la letra con `\lyricsto`. La instrucción `\lyricsto` cambia automáticamente al modo `\lyricmode`, por lo que la palabra clave `\lyricmode` se puede omitir.

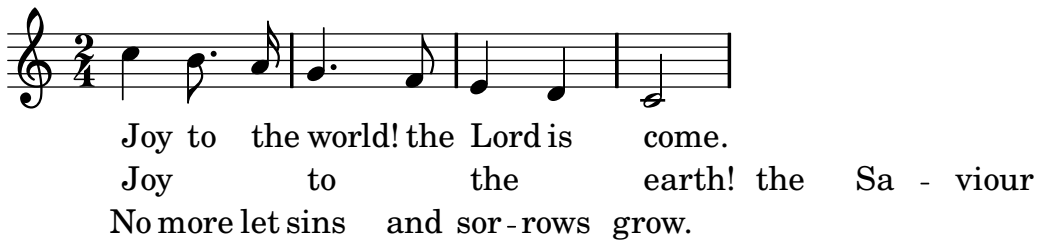
El ejemplo siguiente utiliza comandos diferentes para introducir la letra.

```
<<
  \new Voice = "uno" \relative c'' {
    \autoBeamOff
    \time 2/4
    c4 b8. a16 g4. f8 e4 d c2
  }

% no recomendado: alinea las sílabas a la izquierda
  \new Lyrics \lyricmode { Joy4 to8. the16 world!4. the8 Lord4 is come.2 }

% incorrecto: se necesitan las duraciones
  \new Lyrics \lyricmode { Joy to the earth! the Sa -- viour reigns. }

%correcto
  \new Lyrics \lyricsto "uno" { No more let sins and sor -- rows grow. }
>>
```



El segundo verso no está bien alineado porque las duraciones no se han especificado. Una solución para eso sería usar `\lyricsto`.

El comando `\addlyrics` es realmente tan sólo una forma cómoda de escribir una estructura de LilyPond más complicada que establece la letra.

```
{ MÚSICA }
\addlyrics { LETRA }
```

es lo mismo que

```
\new Voice = "blabla" { MÚSICA }
\new Lyrics \lyricsto "blabla" { LETRA }
```

Duración manual de las sílabas

El texto también se puede introducir sin `\addlyrics` ni `\lyricsto`. En este caso, las sílabas se escriben como notas (pero con las alturas sustituidas por texto) y la duración de cada sílaba se debe escribir explícitamente. Por ejemplo:

```
play2 the4 game2.
sink2 or4 swim2.
```
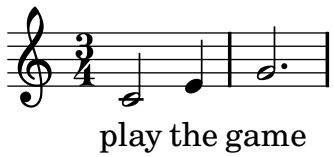
La alineación respecto de una melodía se puede especificar con la propiedad `associatedVoice`,

```
\set associatedVoice = #"lala"
```

El valor de la propiedad (aquí: `"lala"`) ha de ser el nombre de un contexto de Sección “Voice” in *Referencia de Funcionamiento Interno*. Sin este ajuste, las líneas de extensión no se formatearán correctamente.

Presentamos a continuación un ejemplo que muestra el uso de las duraciones manuales de las sílabas del texto:

```
<< \new Voice = "melodia" {
    \time 3/4
    c2 e4 g2.
 }
 \new Lyrics \lyricmode {
   \set associatedVoice = #"melodia"
   play2 the4 game2.
 } >>
```



Véase también

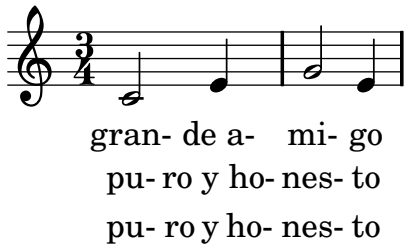
Referencia de funcionamiento interno: Sección “Lyrics” in *Referencia de Funcionamiento Interno*.

Varias sílabas sobre una nota

Para asignar más de una sílaba a una sola nota, puede encerrarlas entre comillas o utilizar un guión bajo (_), para dejar espacios entre las sílabas, o usar el símbolo de tilde curva (~) para obtener una ligadura en el texto[1].

```
\time 3/4
\relative c' { c2 e4 g2 e4 }
\addlyrics { gran- de_a- mi- go }
\addlyrics { pu- "ro y ho-" nes- to }
\addlyrics { pu- ro~y~ho- nes- to }
```

[1] Las ligaduras de texto están hechas con el carácter de Unicode U+203F, por tanto debe asegurarse de tener instalada una tipografía (como DejaVuLGC) que incluya este glifo.



Véase también

Referencia de funcionamiento interno: Sección “LyricCombineMusic” in *Referencia de Funcionamiento Interno*.

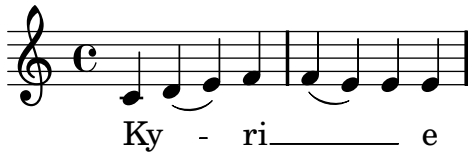
Varias notas sobre una sílaba

A veces, y sobre todo en la música medieval, varias notas se cantan sobre una sílaba única; tales vocalizaciones reciben el nombre de melismas, o melismata.

Podemos definir los melismas completamente dentro de la letra, escribiendo un guión bajo `_` por cada nota que se quiere añadir al melisma.

Además, podemos hacer que se imprima una línea extensora para indicar el melisma en la partitura, escribiendo un doble guión bajo junto a la primera sílaba del melisma. Este ejemplo presenta los tres elementos que se usan con este fin (todos ellos circundados por espacios): guiones dobles para separar las sílabas de una palabra, guiones bajos para añadir notas al melisma, y un doble guión bajo para poner una línea extensora.
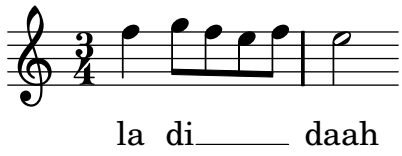
```
{ \set melismaBusyProperties = #'()
 c d( e) f f( e) e e  }
\addlyrics
 { Ky -- _ _ ri __ _ _ _  e }
```



En este caso también podemos tener ligaduras de unión y de expresión en la melodía, si establecemos `melismaBusyProperties`, como se hizo en el ejemplo anterior.

Sin embargo la instrucción `\lyricsto` también puede detectar los melismas automáticamente: sólo pone una sílaba debajo de un grupo de notas unidas mediante ligadura de unión o de expresión. Si quiere forzar que un grupo de notas sin ligadura sea un melisma, inserte `\melisma` después de la primera nota del grupo, y `\melismaEnd` después de la última, p.ej.:

```
<<
  \new Voice = "lala" {
    \time 3/4
    f4 g8
    \melisma
    f e f
    \melismaEnd
    e2
  }
  \new Lyrics \lyricsto "lala" {
    la di __ daah
  }
>>
```



Además, las notas se consideran un melisma si están unidas manualmente mediante una barra, y el barrado automático (véase [Establecer el comportamiento de las barras automáticas], página 58) está desactivado.

Hay un ejemplo completo de una disposición de partitura SATB en la sección Sección “Vocal ensembles” in *Manual de Aprendizaje*.

Instrucciones predefinidas

`\melisma`, `\melismaEnd`.

Véase también

Advertencias y problemas conocidos

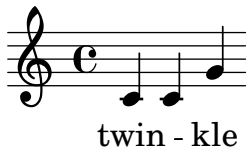
Los melismas no se detectan automáticamente, y las líneas de extensión se deben insertar a mano.

Saltar notas

Se puede conseguir que una línea de letra progrese más lentamente que la melodía, mediante la inserción de desplazamientos o `\skip`s en la letra. Por cada `\skip`, el texto se retrasa en una nota. La instrucción `\skip` debe ir seguida de una duración válida, pero esta duración se ignora cuando se usa `\skip` en las letras.

Por ejemplo,

```
\relative c' { c c g' }
\addlyrics {
  twin -- \skip 4
  kle
}
```



Líneas de extensión y guiones

Los melismas se indican mediante una línea horizontal centrada entre una sílaba y la siguiente. Esta línea recibe el nombre de línea de extensión, y se escribe como ‘ __ ’ (fíjese en los espacios antes y después de los dos guiones bajos).

Nota: Los melismas se indican en la partitura con líneas extensoras, que se escriben con un doble guión bajo; pero también se pueden introducir melismas cortos mediante saltos de notas individuales, que se escriben como caracteres de guión bajo sueltos; de forma predeterminada éstos no producen la impresión de una línea de extensión.

Los guiones centrados se escriben como ‘ -- ’ entre sílabas de una misma palabra (fíjese en los espacios antes y después de los dos guiones). El guión quedará centrado entre las sílabas, y su longitud se ajustará en función del espacio que exista entre ellas.

En la música grabada de modo muy apretado, se pueden quitar los guiones. Se puede controlar si esto ocurre o no, mediante la `minimum-distance` (distancia mínima entre las dos sílabas) y la `minimum-length` (umbral por debajo del cual se suprimen los guiones).

Véase también

Referencia de funcionamiento interno: Sección "LyricExtender" in *Referencia de Funcionamiento Interno*, Sección "LyricHyphen" in *Referencia de Funcionamiento Interno*.

Letras y repeticiones

Continuará...
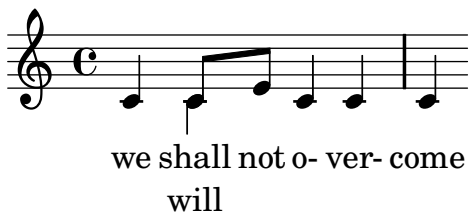
2.1.4 Usos específicos de la letra

A menudo se aplican a una melodía distintos versos de una canción de formas ligeramente distintas. Tales variantes pueden aún captarse mediante `\lyricsto`.

Letras en divisi

Puede presentar letras alternativas (o divisi) nombrando contextos de voz y aplicando la letra a los mencionados contextos específicos.
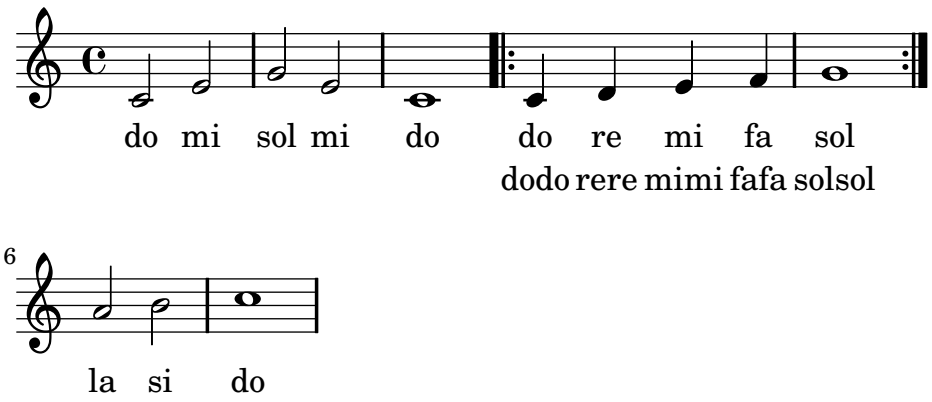
```
\score{ <<
  \new Voice = "melodia" {
    \relative c' {
      c4
      <<
        { \voiceOne c8 e }
        \new Voice = "partedividida" { \voiceTwo c4 }
      >>
      \oneVoice c4 c | c
    }
  }
  \new Lyrics \lyricsto "melodia" { we shall not o- ver- come }
  \new Lyrics \lyricsto "partedividida" { will }
>> }
```



Puede usar este truco para presentar las distintas letras de una sección que se repite.

```
\score{ <<
  \new Voice = "melodia" \relative c' {
    c2 e | g e | c1 |
    \new Voice = "estrofa" \repeat volta 2 {c4 d e f | g1 | }
    a2 b | c1}
  \new Lyrics = "mainlyrics" \lyricsto melodia \lyricmode {
    do mi sol mi do
    la si do }
  \context Lyrics = "mainlyrics" \lyricsto estrofa \lyricmode {
   do re mi fa sol }
```

```
  \new Lyrics = "repeatlyrics" \lyricsto estrofa \lyricmode {
   dodo rere mimi fafa solsol }
>>
}
```
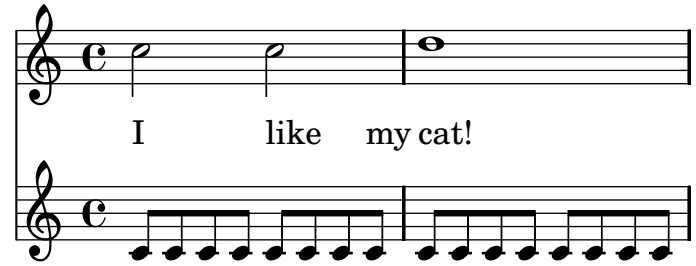


Letra independiente de las notas

En cierta música vocal compleja, podría ser deseable colocar la letra de forma totalmente independiente de las notas. La música que se define dentro de `lyricrhythm` desaparece dentro del contexto `Devnull`, pero los valores rítmicos aún se pueden utilizar para colocar la letra.

```
voz = {
  c''2
  \tag #'music { c''2 }
  \tag #'lyricrhythm { c''4. c''8 }
  d''1
}

letra = \lyricmode { I like my cat! }

<<
  \new Staff \keepWithTag #'music \voz
  \new Devnull="nowhere" \keepWithTag #'lyricrhythm \voz
  \new Lyrics \lyricsto "nowhere" \letra
  \new Staff { c'8 c' c' c' c' c' c' c'
  c' c' c' c' c' c' c' c' }
>>
```

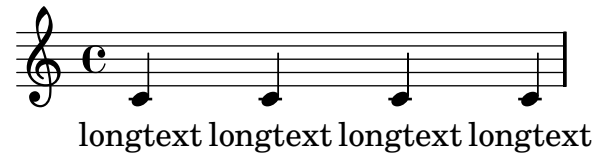


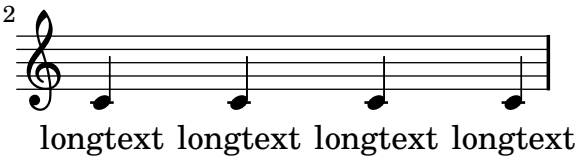
Sólo se recomienda este método si la música que está en el contexto `Devnull` no contiene melismas. Los melismas están definidos por el contexto `Voice`. La conexión de la letra a un contexto `Devnull` hace que se pierdan los enlaces entre la voz y las sílabas, y también la información relativa a los melismas. Por tanto, si enlazamos la letra a un contexto `Devnull`, los melismas implícitos se ignoran.

Espaciado de las sílabas

Para aumentar el espaciado entre las sílabas de la letra, establezca la propiedad minimum-distance de LyricSpace.
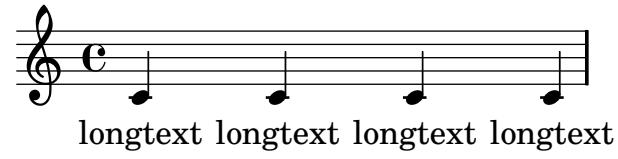
```
{
  c c c c
  \override Lyrics.LyricSpace #'minimum-distance = #1.0
  c c c c
}
\addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
}
```
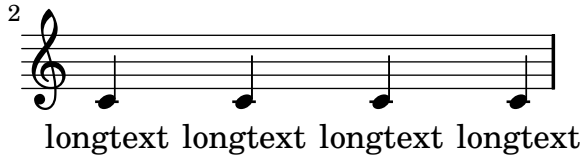




Para que este cambio sea válido para la letra de la partitura completa, establezca la propiedad dentro del 'layout' o disposición.

```
\score {
  \relative c' {
  c c c c
  c c c c
  }
  \addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
  }
  \layout {
    \context {
      \Lyrics
      \override LyricSpace #'minimum-distance = #1.0
    }
  }
}
```





Fragmentos de código seleccionados

La comprobación que asegura que las inscripciones de texto y la letra de las canciones se mantienen dentro de los márgenes, es una tarea computacional relativamente costosa. Para acelerar el procesado, LilyPond no hace tales cálculos de forma predeterminada; para habilitarlos, utilice

```
\override Score.PaperColumn #'keep-inside-line = ##t
```

Para hacer también que las sílabas de la letra eviten las barras de compás, use

```
\layout {
  \context {
    \Lyrics
      \consists "Bar_engraver"
      \consists "Separating_line_group_engraver"
      \override BarLine #'transparent = ##t
  }
}
```

Centrado de la letra entre pentagramas

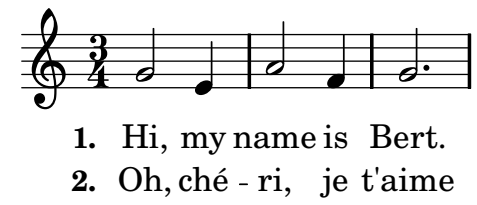
Continuará...

2.1.5 Versos

Añadir números de verso

Los números de los versos se pueden añadir estableciendo `stanza`, p.ej.,

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set stanza = #"1. "
  Hi, my name is Bert.
} \addlyrics {
  \set stanza = #"2. "
  Oh, ché -- ri, je t'aime
}
```
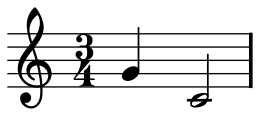


Estos números aparecerán inmediatamente antes de la primera sílaba.

Añadir expresiones dinámicas a los versos

Los versos que difieren en su sonoridad se pueden especificar escribiendo una indicación dinámica antes de cada verso. En LilyPond, todo lo que aparece delante de un verso está dentro del objeto `StanzaNumber`; las indicaciones dinámicas no son diferentes. Por razones técnicas, tendrá que establecer el valor de la sección stanza (verso) fuera de `\lyricmode`:

```
texto = {
  \set stanza = \markup { \dynamic "ff" "1. " }
  \lyricmode {
    Big bang
  }
}

<<
  \new Voice = "melodia" {
    \time 3/4
    g'4 c'2
  }
\new Lyrics \lyricsto "melodia" \texto
>>
```
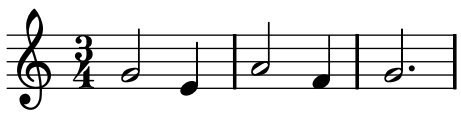


ff **1.** Big bang

Añadir el nombre de los cantantes a los versos

También se pueden poner los nombres de los cantantes. Se imprimen al comienzo de la línea, igual que los nombres de instrumento. Se crean estableciendo un valor para `vocalName`. Se puede definir una versión abreviada como `shortVocalName`.

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set vocalName = #"Bert "
  Hi, my name is Bert.
} \addlyrics {
  \set vocalName = #"Ernie "
  Oh, ché -- ri, je t'aime
}
```



Bert Hi, my name is Bert.
Ernie Oh, ché - ri, je t'aime

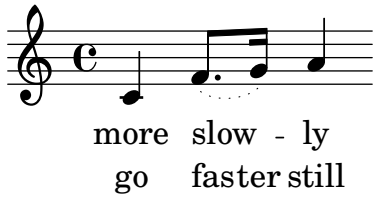
Versos con ritmos distintos

Ignorar los melismas

Existe la posibilidad de que el texto tenga un melisma en un verso, pero varias sílabas en otro. Una solución es hacer que la voz más rápida ignore el melisma. Esto se consigue estableciendo `ignoreMelismata` en el contexto Lyrics.

```
<<
  \relative c' \new Voice = "lala" {
    \set Staff.autoBeaming = ##f
    c4
```

```
    \slurDotted
    f8.[( g16])
    a4
  }
  \new Lyrics \lyricsto "lala" {
    more slow -- ly
  }
  \new Lyrics \lyricsto "lala" {
    go
    \set ignoreMelismata = ##t
    fas -- ter
    \unset ignoreMelismata
    still
  }
>>
```

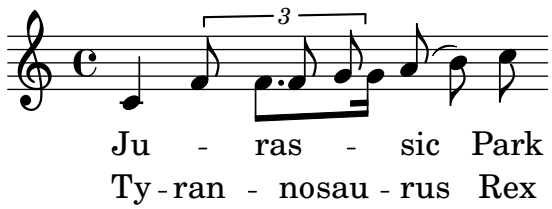


Advertencias y problemas conocidos

A diferencia de casi todas las instrucciones `\set`, `\set ignoreMelismata` no funciona si va precedido de `\once`. Es necesario utilizar `\set` y `\unset` para delimitar la letra en que se quieren ignorar los melismas.

Cambio a una melodía alternativa

Son posibles variaciones más complejas en la disposición del texto. Se puede cambiar la melodía para una línea de la letra durante el texto. Esto se hace estableciendo la propiedad `associatedVoice`. En el ejemplo



el texto del primer verso se aplica a una melodía llamada 'lahlah',

```
\new Lyrics \lyricsto "lahlah" {
  Ju -- ras -- sic Park
}
```

El segundo verso se aplica en principio al contexto de `lahlah`, pero para la sílaba 'ran', cambia a una melodía diferente. Esto se consigue con

```
\set associatedVoice = alternative
```

Aquí, `alternative` es el nombre del contexto de `Voice` que contiene el tresillo.

La instrucción debe ir una sílaba antes de la cuenta, antes de 'Ty' en este caso. En otras palabras, el cambio en la voz asociada associatedVoice se produce un paso más tarde de lo que se espera. Esto es así por razones técnicas, y no es un fallo del programa.

```
\new Lyrics \lyricsto "lahlah" {
  \set associatedVoice = alternative % se aplica al "ran"
  Ty --
  ran --
  no --
  \set associatedVoice = lahlah % se aplica al "rus"
  sau -- rus Rex
}
```

La disposición se vuelve a cambiar a la situación inicial mediante la asignación de `lahlah` a `associatedVoice`.

Imprimir los versos al final

En ocasiones es conveniente tener un verso ajustado a la música, y el resto añadido en forma de estrofa al final de la pieza. Esto se puede conseguir escribiendo los versos adicionales dentro de una sección `\markup` fuera del bloque score principal de la partitura. Tenga en cuenta que existen dos formas distintas de forzar los saltos de línea al utilizar `\markup`.

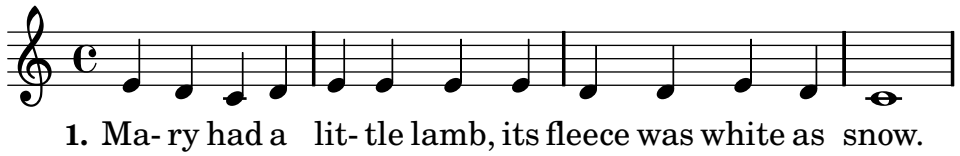
```
melodia = \relative c' {
e d c d | e e e e |
d d e d | c1 |
}

texto = \lyricmode {
\set stanza = #"1." Ma- ry had a lit- tle lamb,
its fleece was white as snow.
}

\score{ <<
  \new Voice = "uno" { \melodia }
  \new Lyrics \lyricsto "uno" \texto
>>
  \layout { }
}
\markup { \column{
  \line{ Verse 2. }
  \line{ All the children laughed and played }
  \line{ To see a lamb at school. }
  }
}
\markup{
  \wordwrap-string #"
  Verse 3.

  Mary took it home again,

  It was against the rule."
}
```



Verse 2.
All the children laughed and played
To see a lamb at school.

Verse 3.
Mary took it home again,
It was against the rule.

Imprimir los versos al final en varias columnas

Cuando una pieza tiene muchos versos, a menudo se imprimen en varias columnas a lo largo de toda la página. Con frecuencia un número de verso fuera del margen precede a cada verso. El ejemplo siguiente muestra cómo producir dicha salida en Lilypond.
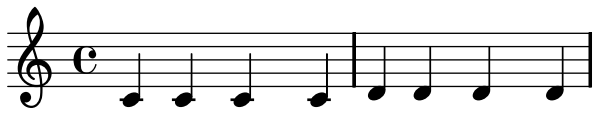
```
melodia = \relative c' {
  c c c c | d d d d
}

texto = \lyricmode {
  \set stanza = #"1." This is verse one.
  It has two lines.
}

\score{ <<
    \new Voice = "uno" { \melodia }
    \new Lyrics \lyricsto "uno" \texto
   >>
  \layout { }
}

\markup {
  \fill-line {
    \hspace #0.1 % aleja la columna del margen izquierdo;
        % se puede quitar si el espaciado de la página es apretado
     \column {
      \line { \bold "2."
        \column {
          "This is verse two."
          "It has two lines."
        }
      }
      \hspace #0.1 % añade espaciado vertical entre estrofas
      \line { \bold "3."
        \column {
          "This is verse three."
          "It has two lines."
        }
      }
    }
    \hspace #0.1  % añade espaciado horizontal entre columnas;
        % if they are still too close, add more " " pairs
        % hasta que el resultado resulte adecuado
     \column {
      \line { \bold "4."
        \column {
```

```
          "This is verse four."
          "It has two lines."
        }
      }
      \hspace #0.1 % añade espaciado vertical entre estrofas
      \line { \bold "5."
        \column {
          "This is verse five."
          "It has two lines."
        }
      }
    }
  \hspace #0.1 % aplica más separación en el margen derecho;
      % se puede quitar si el espaciado de la página es apretado
  }
}
```



1. This is verse one. It has two lines.

2. This is verse two.
It has two lines.

3. This is verse three.
It has two lines.

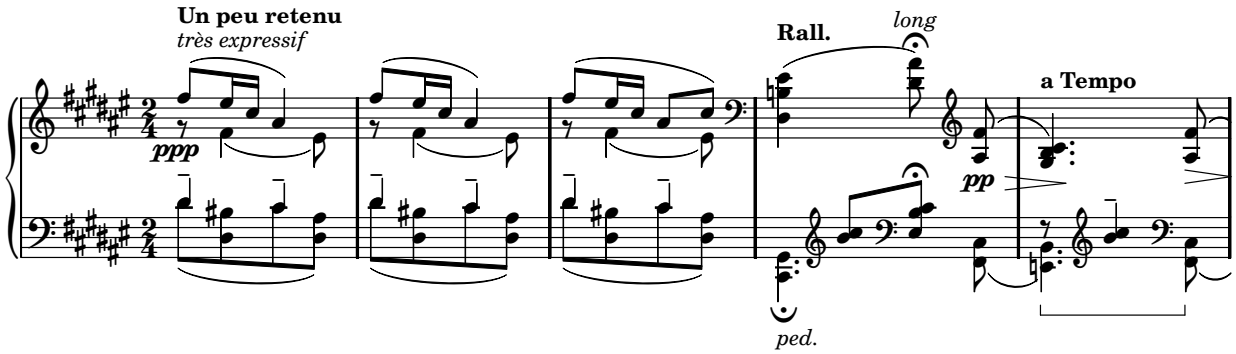
4. This is verse four.
It has two lines.
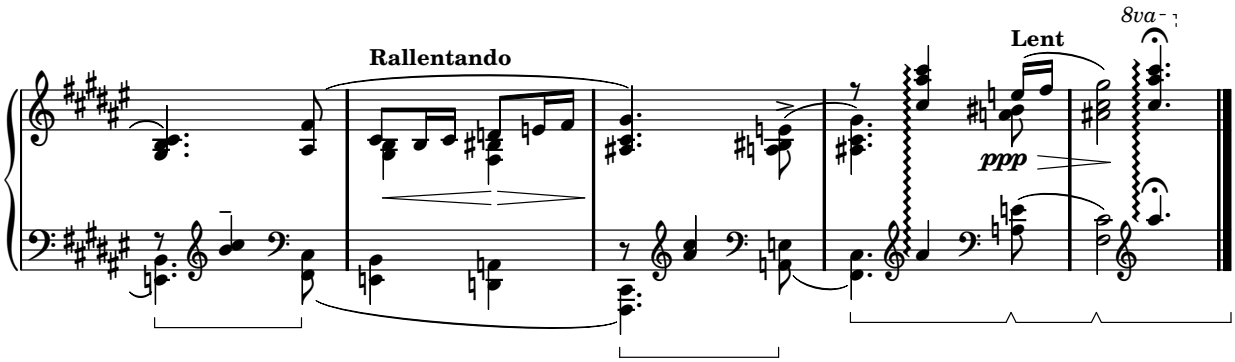
5. This is verse five.
It has two lines.

Véase también

Referencia de funcionamiento interno: Sección “LyricText” in *Referencia de Funcionamiento Interno*, Sección “StanzaNumber” in *Referencia de Funcionamiento Interno*.

2.2 Teclados y otros instrumentos de varios pentagramas





Esta sección se ocupa de varios aspectos de la notación musical que son exclusivos de los instrumentos de teclado y otros instrumentos cuya notación se realiza sobre varios pentagramas, como el arpa o el vibráfono. A los efectos de nomenclatura, en esta sección se denominan abreviadamente "teclados" a todo este grupo de instrumentos de varios pentagramas, aunque algunos de ellos no tienen teclado.

2.2.1 Notación común para instrumentos de teclado

Esta sección trata aspectos de notación que pueden aparecer en casi todos los instrumentos de teclado.

Referencias para teclados

La notación de los instrumentos de teclado se suele realizar con sistemas de piano. Éstos son dos pentagramas normales agrupados mediante una llave. También se utiliza esta notación para otros instrumentos de teclado. La música de órgano se escribe normalmente con dos pentagramas dentro de un grupo `PianoStaff` y un tercer pentagrama normal para los pedales.

Los pentagramas son hasta cierto punto independientes, pero a veces las voces pueden cruzarse entre los dos pentagramas. Esta sección trata técnicas de notación que son particulares de la música de teclado.

En otros lugares se tratan varios problemas comunes de la música de teclado:

- La música para teclado suele contener varias voces y el número de éstas puede variar ampliamente; esto se describe en [Resolución de las colisiones], página 116.
- La música para teclado se puede escribir en paralelo, como se describe en [Escribir música en paralelo], página 123.
- Las digitaciones se indican como puede verse en [Indicaciones de digitación], página 155.
- Las indicaciones de pedal de órgano se insertan como articulaciones, véase Sección B.10 [Lista de articulaciones], página 513.
- Las líneas verticales de rejilla se pueden mostrar como se describe en [Líneas de rejilla], página 162.
- La música para teclado incluye con frecuencia ligaduras *Laissez vibrer* así como ligaduras sobre arpegios y trémolos, descritos en [Ligaduras de unión], página 36.
- La colocación de arpegios a través de varias voces y pentagramas se cubre en [Arpegio], página 97.
- Las indicaciones de trémolo se describen en [Repeticiones de trémolo], página 110.
- Varios de los trucos que pueden ser necesarios en música para teclado se muestran en Sección "Real music example" in *Manual de Aprendizaje*.
- Las notas ocultas se pueden usar para producir ligaduras de unión que se cruzan entre voces, como se explica en Sección "Other uses for tweaks" in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección “Real music example” in *Manual de Aprendizaje*, Sección “Other uses for tweaks” in *Manual de Aprendizaje*.

Referencia de la notación: [Nombres de instrumentos], página 145, [Resolución de las colisiones], página 116, [Escribir música en paralelo], página 123, [Indicaciones de digitación], página 155, Sección B.10 [Lista de articulaciones], página 513, [Líneas de rejilla], página 162, [Ligaduras de unión], página 36, [Arpegio], página 97, [Repeticiones de trémolo], página 110.

Referencia de funcionamiento interno: Sección “PianoStaff” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Advertencias y problemas conocidos

Los matices no se colocan centrados, pero se pueden utilizar ciertas alternativas. Una opción es la plantilla ‘piano con matices centrados’ que está en Sección “Piano templates” in *Manual de Aprendizaje*; otra opción es aumentar el relleno (`staff-padding`) de los matices como se estudia en Sección “Moving objects” in *Manual de Aprendizaje*.

Cambiar de pentagrama manualmente

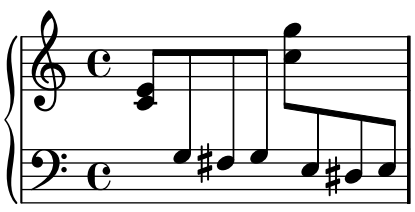
Las voces se pueden cambiar de pentagrama manualmente usando la instrucción

`\change Staff = nombre_del_pentagrama`

La cadena *nombre_del_pentagrama* es el nombre del pentagrama. Cambia la voz actual de su pentagrama en curso al pentagrama de nombre *nombre_del_pentagrama*. Los valores más corrientes para *nombre_del_pentagrama* son `"arriba"` y `"abajo"`, o bien `"MD"` y `"MI"`.

Las notas de pentagrama cruzado reciben la barra automáticamente:

```
\new PianoStaff <<
  \new Staff = "arriba" {
    <e' c'>8
    \change Staff = "abajo"
    g8 fis g
    \change Staff = "arriba"
    <g'' c''>8
    \change Staff = "abajo"
    e8 dis e
    \change Staff = "arriba"
  }
  \new Staff = "abajo" {
    \clef bass
    % mantener vivo el pentagrama
    s1
  }
>>
```



Si el barrado necesita algún trucaje, efectúe en primer lugar los cambios en las direcciones de las plicas. Las posiciones de las barras se miden posteriormente a partir del centro del pentagrama que está más cerca de la barra. Para ver un ejemplo sencillo de trucaje de las barras, consulte Sección “Fixing overlapping notation” in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección “Fixing overlapping notation” in *Manual de Aprendizaje*.

Referencia de la notación: [Plicas], página 160, [Barras automáticas], página 56.

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Beam” in *Referencia de Funcionamiento Interno*, Sección “ContextChange” in *Referencia de Funcionamiento Interno*.
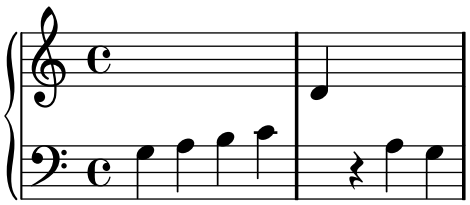
Cambiar de pentagrama automáticamente

Se puede hacer que las voces cambien automáticamente entre los pentagramas inferior y superior. La sintaxis para conseguir esto es

`\autochange ...`*`música`*`...`

Esto creará dos pentagramas dentro del grupo actual de pentagramas (normalmente un `PianoStaff`), llamados `"up"` (arriba) y `"down"` (abajo). El pentagrama inferior estará en clave de Fa de forma predeterminada. El cambiador automático hace el cambio basándose en la altura (el Do central es el punto de inflexión), y se adelanta saltando sobre los silencios para hacer cambios por adelantado.

```
\new PianoStaff {
  \autochange {
    g4 a b c'
    d'4 r a g
  }
}
```



Una sección marcada `\relative` que se encuentra fuera de `\autochange` no tiene efecto sobre las notas de la música; por tanto, en caso necesario escriba `\relative` dentro de `\autochange`.

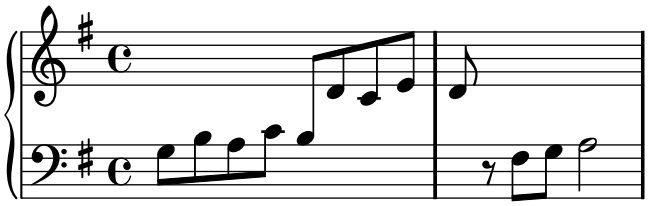
Si se necesita más control sobre los pentagramas individuales, se pueden crear manualmente con los nombres `"up"` y `"down"`. Entonces, la instrucción `\autochange` alternará su voz entre los pentagramas existentes.

Nota: Si se crean los pentagramas manualmente, *se deben* llamar `"up"` y `"down"` (en inglés).

Por ejemplo, esto es necesario para colocar una armadura de tonalidad en el pentagrama inferior:

```
\new PianoStaff <<
  \new Staff = "up" {
    \new Voice = "melodiaUno" {
      \key g \major
```

```
      \autochange \relative c' {
        g8 b a c b d c e
        d8 r fis, g a2
      }
    }
  }
  \new Staff = "down" {
    \key g \major
    \clef bass
  }
>>
```



Véase también

Referencia de la notación: [Cambiar de pentagrama manualmente], página 209.

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “AutoChangeMusic” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

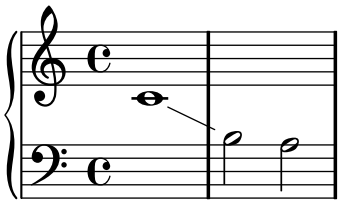
Los cambios de pentagrama pueden no acabar en lugares óptimos. Para conseguir una salida de alta calidad es mejor especificar manualmente los cambios de pentagrama.

Los acordes no se dividen entre los pentagramas; se asignan a un pentagrama en función de la primera nota nombrada dentro de la construcción del acorde.

Líneas de cambio de pentagrama

Cuando una voz cambia a otro pentagrama, se puede imprimir automáticamente una línea que conecte las notas:

```
\new PianoStaff <<
  \new Staff = "uno" {
    \showStaffSwitch
    c1
    \change Staff = "dos"
    b2 a
  }
  \new Staff = "dos" {
    \clef bass
    s1*2
  }
>>
```



Instrucciones predefinidas

`\showStaffSwitch`, `\hideStaffSwitch`.

Véase también

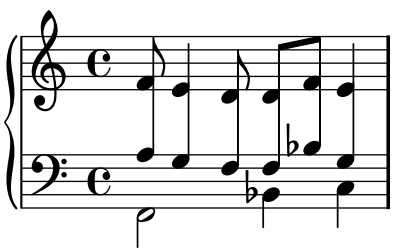
Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_head_line_engraver” in *Referencia de Funcionamiento Interno*, Sección “VoiceFollower” in *Referencia de Funcionamiento Interno*.

Plicas de pentagrama cruzado

Se pueden hacer acordes que cruzan los pentagramas:

```
\new PianoStaff <<
  \new Staff {
    \relative c' {
      f8 e4 d8 d f e4
    }
  }
  \new Staff {
    \relative c' {
      << {
        \clef bass
        % las plicas pueden superponerse al otro pentagrama
        \override Stem #'cross-staff = ##t
        % extender las plicas para que lleguen hasta el otro pentagrama
        \override Stem #'length = #12
        % no imprimir corchetes adicionales
        \override Stem #'flag-style = #'no-flag
        % evitar el barrado según necesidades
        a8 g4 f8 f bes\noBeam g4
      }
      \\
      {
        f,2 bes4 c
      } >>
    }
  }
>>
```



Fragmentos de código seleccionados

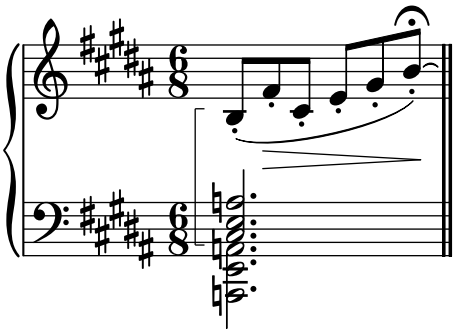
Indicar acordes de pentagrama cruzado con corchetes de arpegio

Un corchete de arpegio puede indicar que se tienen que tocadr con la misma mano notas que están en dos pentagramas distintos. Para hacerlo, el `PianoStaff` se debe configurar para que acepte símbolos de arpegio de pentagrama cruzado y los símbolos de arpegio se deben configurar a la forma de corchete en el contexto de `PianoStaff`.

(Debussy, Les collines d'Anacapri, m. 65)

```
\paper { ragged-right = ##t }

\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \override PianoStaff.Arpeggio #'stencil = #ly:arpeggio::brew-chord-bracket
  \new Staff {
    \relative c' {
      \key b \major
      \time 6/8
      b8-.(\arpeggio fis'-.\> cis-. e-. gis-. b-.)\!\fermata^\laissezVibrer
      \bar "||"
    }
  }
  \new Staff {
    \relative c' {
      \clef bass
      \key b \major
      <<
        {
          <a e cis>2.\arpeggio
        }
        \\
        {
          <a, e a,>2.
        }
      >>
    }
  }
>>
```



Véase también

Fragmentos de código: Sección "Keyboards" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Stem” in *Referencia de Funcionamiento Interno*.
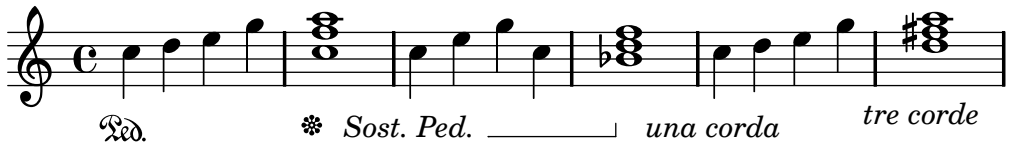
2.2.2 Piano

Esta sección trata de aspectos de la notación directamente relacionados con el piano.

Pedales de piano
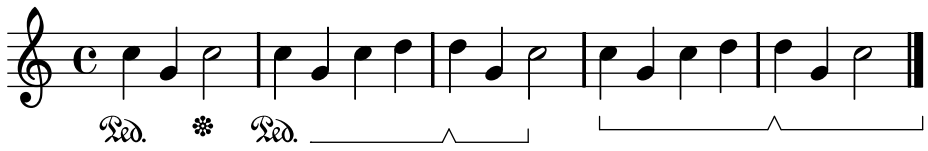
En general, los pianos tienen tres pedales que alteran la forma de producir el sonido: el pedal de mantenimiento (pedal *derecho* o pedal *forte*), el pedal *sostenuto* (*sos.*, pedal central) y el pedal *una corda* (*U.C.*, pedal izquierdo). Los pedales de mantenimiento como el pedal derecho del piano se encuentran también en el vibráfono y en la celesta.

```
c4\sustainOn d e g
<c, f a>1\sustainOff
c4\sostenutoOn e g c,
<bes d f>1\sostenutoOff
c4\unaCorda d e g
<d fis a>1\treCorde
```



Existen tres estilos de indicaciones de pedal: textual, corchete y mixta. El pedal derecho y el pedal de una corda utilizan el estilo textual de forma predeterminada, mientras que el pedal sostenuto utiliza el método mixto de forma predeterminada.

```
c4\sustainOn g c2\sustainOff
\set Staff.pedalSustainStyle = #'mixed
c4\sustainOn g c d
d\sustainOff\sustainOn g, c2\sustainOff
\set Staff.pedalSustainStyle = #'bracket
c4\sustainOn g c d
d\sustainOff\sustainOn g, c2
\bar "|."
```



La colocación de las instrucciones de pedal se corresponden con el movimiento físico del pedal derecho en el transcurso de la ejecución. La pedalización hasta la barra final de compás se indica mediante la omisión de la instrucción final de levantar el pedal.

Véase también

Referencia de la notación: [Ligaduras de unión], página 36.

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SustainPedal” in *Referencia de Funcionamiento Interno*, Sección “SustainPedalLineSpanner” in *Referencia de Funcionamiento Interno*, Sección “SustainEvent” in *Referencia de Funcionamiento Interno*, Sección “SostenutoPedal” in *Referencia de Funcionamiento Interno*, Sección “SostenutoPedalLineSpanner”

in *Referencia de Funcionamiento Interno*, Sección "SostenutoEvent" in *Referencia de Funcionamiento Interno*, Sección "UnaCordaPedal" in *Referencia de Funcionamiento Interno*, Sección "UnaCordaPedalLineSpanner" in *Referencia de Funcionamiento Interno*, Sección "UnaCordaEvent" in *Referencia de Funcionamiento Interno*, Sección "PianoPedalBracket" in *Referencia de Funcionamiento Interno*, Sección "Piano_pedal_engraver" in *Referencia de Funcionamiento Interno*.

2.2.3 Acordeón

Esta sección se ocupa de la notación que es exclusiva del acordeón.

Símbolos de Discanto

Los acordeones se suelen construir con más de un conjunto de lengüetas que pueden estar al unísono, una octava por encima, o una octava por debajo de la altura escrita. Cada fabricante de acordeones utiliza distintos nombres para los *cambios* que seleccionan las distintas combinaciones de lengüetas, como *oboe*, *musette* o *bandoneón*, de manera que se ha venido en usar un sistema de signos para simplificar las instrucciones de ejecución.

Fragmentos de código seleccionados

Símbolos de acordeón discanto

Los símbolos específicos de acordeón discanto se escriben mediante `\markup`. Se puede trucar la colocación vertical de los símbolos modificando los argumentos de `\raise`.

```
discanto = \markup {
  \musicglyph #"accordion.accDiscant"
}
punto = \markup {
  \musicglyph #"accordion.accDot"
}

\layout { ragged-right = ##t }

% registro de 16 pies
acordFagot = ^\markup {
  \combine
  \discanto
  \raise #0.5 \punto
}

% registro de een korig 8 en 16 pies
acordBandon = ^\markup {
  \combine
    \discanto
    \combine
      \raise #0.5 \punto
      \raise #1.5 \punto
}

acordVCello = ^\markup {
  \combine
    \discanto
    \combine
      \raise #0.5 \punto
```

```
      \combine
        \raise #1.5 \punto
        \translate #'(1 . 0) \raise #1.5 \punto
}

% registro de 4-8-16 pies
acordArmon = ^\markup {
  \combine
    \discanto
    \combine
      \raise #0.5 \punto
      \combine
        \raise #1.5 \punto
        \raise #2.5 \punto
}

acordTrombon = ^\markup {
  \combine
    \discanto
    \combine
      \raise #0.5 \punto
      \combine
        \raise #1.5 \punto
        \combine
          \translate #'(1 . 0) \raise #1.5 \punto
          \translate #'(-1 . 0) \raise #1.5 \punto
}

% registro de eenkorig 4 y 16 pies
acordOrgano = ^\markup {
  \combine
    \discanto
    \combine
      \raise #0.5 \punto
      \raise #2.5 \punto
}

acordMaestro = ^\markup {
  \combine
    \discanto
    \combine
      \raise #0.5 \punto
      \combine
        \raise #1.5 \punto
        \combine
          \translate #'(1 . 0) \raise #1.5 \punto
          \combine
            \translate #'(-1 . 0) \raise #1.5 \punto
            \raise #2.5 \punto
}

acordAcordeon = ^\markup {
```

```
  \combine
    \discanto
    \combine
      \raise #1.5 \punto
      \combine
        \translate #'(1 . 0) \raise #1.5 \punto
        \combine
          \translate #'(-1 . 0) \raise #1.5 \punto
          \raise #2.5 \punto
}

accordMusette = ^\markup {
  \combine
    \discanto
    \combine
      \raise #1.5 \punto
      \combine
        \translate #'(1 . 0) \raise #1.5 \punto
        \translate #'(-1 . 0) \raise #1.5 \punto
}

acordCeleste = ^\markup {
  \combine
    \discanto
    \combine
      \raise #1.5 \punto
      \translate #'(-1 . 0) \raise #1.5 \punto
}

acordOboe = ^\markup {
  \combine
    \discanto
    \combine
      \raise #1.5 \punto
      \raise #2.5 \punto
}

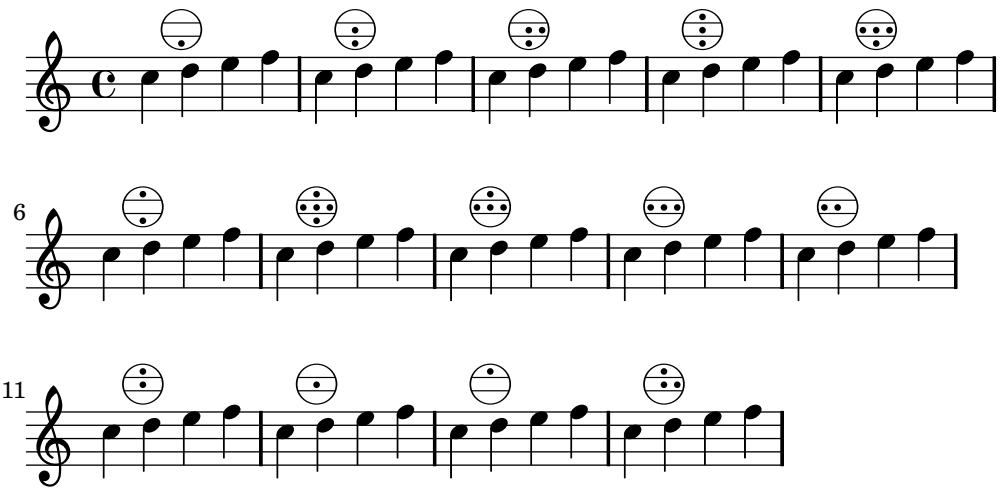
acordClarin = ^\markup {
  \combine
    \discanto
    \raise #1.5 \punto
}

acordPiccolo = ^\markup {
    \combine
       \discanto
       \raise #2.5 \punto
}

acordViolin = ^\markup {
  \combine
    \discanto
```

```
    \combine
      \raise #1.5 \punto
      \combine
        \translate #'(1 . 0) \raise #1.5 \punto
        \raise #2.5 \punto
}

\relative c'' {
  c4 d\acordFagot e f
  c4 d\acordBandon e f
  c4 d\acordVCello e f
  c4 d\acordArmon e f
  c4 d\acordTrombon e f
  \break
  c4 d\acordOrgano e f
  c4 d\acordMaestro e f
  c4 d\acordAcordeon e f
  c4 d\accordMusette e f
  c4 d\acordCeleste e f
  \break
  c4 d\acordOboe e f
  c4 d\acordClarin e f
  c4 d\acordPiccolo e f
  c4 d\acordViolin e f
}
```



Véase también

Fragmentos de código: Sección “Keyboards” in *Fragmentos de código*.

2.2.4 Arpa

Esta sección trata sobre asuntos de notación específicos del arpa.

Referencias para notación de arpa

Ciertas características comunes de la música de arpa se estudian en otras partes de la documentación:

- El glissando es la técnica más característica del arpa, [Glissando], página 96.

- El *bisbigliando* se escribe como un trémolo, [Repeticiones de trémolo], página 110
- Los armónicos naturales se estudian bajo el epígrafe [Armónicos], página 221.
- Para los arpegios dirigidos y los no arpegios, véase [Arpegio], página 97.
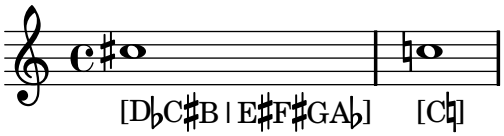
Véase también

Referencia de la notación: [Repeticiones de trémolo], página 110, [Glissando], página 96, [Arpegio], página 97, [Armónicos], página 221.

Pedales de arpa

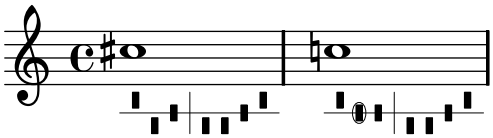
Las arpas tienen siete cuerdas por octava que pueden sonar a la altura natural, bemol o sostenido. En el arpa de palancas o arpa celta, cada cuerda se ajusta individualmente, pero en las arpas de pedales todas las cuerdas con igual nombre de nota se controlan por un único pedal. Visto de izquierda a derecha desde la posición del ejecutante, los pedales son Re, Do y Si a la izquierda y Mi, Fa, Sol y La a la derecha. La posición de los pedales se puede indicar con indicaciones textuales:

```
\textLengthOn
cis1_\markup \concat \vcenter { [D \flat C \sharp B|E \sharp F \sharp G A \flat] }
c!1_\markup \concat \vcenter {[ C \natural ]}
```



o mediante diagramas de pedal:

```
\textLengthOn
cis1_\markup { \harp-pedal #"^v-|vv-^" }
c!1_\markup { \harp-pedal #"^o--|vv-^" }
```

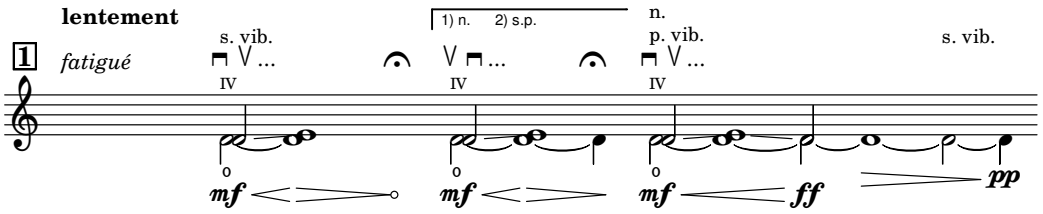


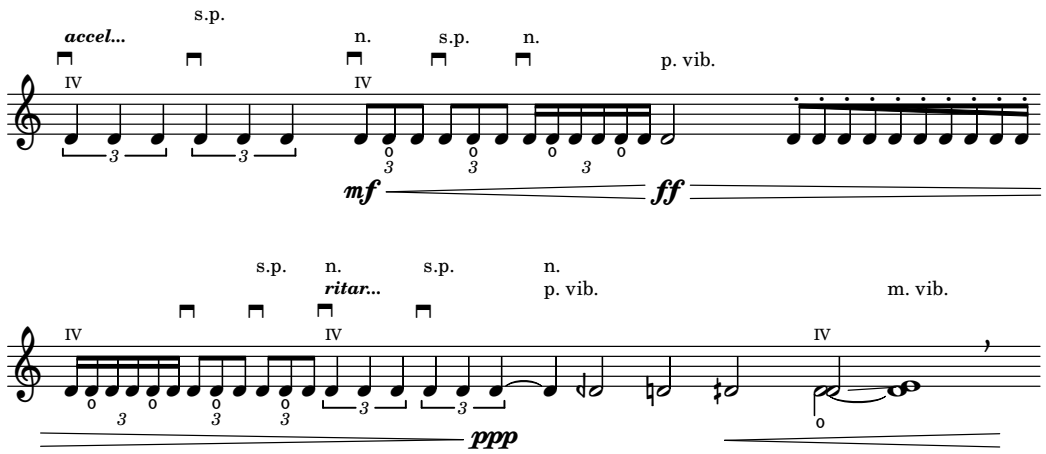
La instrucción `\harp-pedal` acepta una cadena de caracteres en la que `^` es la posición alta del pedal (altura bemol), `-` es la posición intermedia (altura natural), `v` es la posición baja (altura sostenido), y `|` es la línea vertical separadora. Si se antepone `o`, el siguiente símbolo de pedal se rodea por una elipse.

Véase también

Referencia de la notación: [Guiones de texto], página 166, Sección B.8.5 [Instrument Specific Markup], página 505.

2.3 Instrumentos de cuerda sin trastes





Esta sección aporta información y referencias que resultan útiles si se escribe música para instrumentos de cuerda sin trastes, principalmente instrumentos de cuerda orquestales.

2.3.1 Notación común para cuerdas sin trastes

Hay poca notación especializada para los instrumentos de cuerdas sin trastes. La notación de la música se realiza en un solo pentagrama, y se suele usar una sola voz. Se pueden necesitar dos voces para algunos pasages en dobles cuerdas o en divisi.

Referencias para cuerdas sin trastes

La mayor parte de la notación que resulta de utilidad para las cuerdas orquestales y otros instrumentos de arco se estudia en otras partes del manual:

- Las indicaciones textuales como “pizz.” y “arco” se añaden como texto simple: véase [Guiones de texto], página 166.
- Las digitaciones, entre ellas la indicación de pulgar, se describen en [Indicaciones de digitación], página 155.
- Las dobles cuerdas se indican normalmente escribiendo un acorde, véase [Notas en acorde], página 111. Hay otras indicaciones que se pueden añadir para la interpretación de acordes, véase [Arpegio], página 97.
- Hay una plantilla para cuarteto de cuerda en Sección “String quartet” in *Manual de Aprendizaje*. Otras se muestran en la sección de fragmentos de código.

Véase también

Manual de aprendizaje: Sección “String quartet” in *Manual de Aprendizaje*.

Referencia de la notación: [Guiones de texto], página 166, [Indicaciones de digitación], página 155, [Notas en acorde], página 111, [Arpegio], página 97.
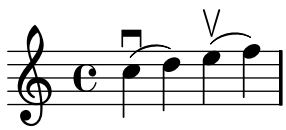
Fragmentos de código: Sección “Unfretted strings” in *Fragmentos de código*.

Indicaciones de arco

Las indicaciones de arco se crean como articulaciones, que se describen en [Articulaciones y ornamentos], página 84.
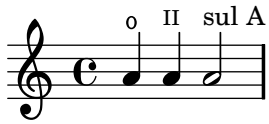
Las instrucciones de arco `\upbow` (arco arriba) y `\downbow` (arco abajo) se usan con ligaduras de expresión de la siguiente manera:

```
c4(\downbow d) e(\upbow f)
```



y el ejemplo siguiente muestra tres formas distintas de indicar un La sobre una cuerda al aire del violín:

```
a4 \open
a^\markup { \teeny "II" }
a2^\markup { \small "sul A" }
```



Instrucciones predefinidas

`\downbow`, `\upbow`, `\open`.

Véase también

Referencia de la notación: [Articulaciones y ornamentos], página 84, [Ligaduras de expresión], página 92.

Armónicos
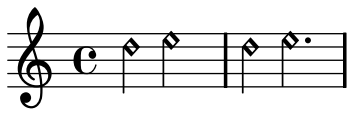
Armónicos naturales

La notación de los armónicos naturales se puede realizar de varias formas. Generalmente, una nota con la cabeza en forma de rombo significa tocar (sin apretar) la cuerda en el lugar donde se pisaría la nota si no fuese un rombo.

> **Nota:** Los armónicos **se deben** definir dentro de una construcción de acorde incluso si hay una sola nota.

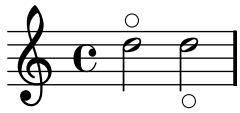
Los armónicos con puntillo indicados con `\harmonic` no muestran el puntillo. Se debe establecer la propiedad de contexto `harmonicDots` si se necesita el puntillo.

```
<d\harmonic>4 <e\harmonic>2.
\set harmonicDots = ##t
<d\harmonic>4 <e\harmonic>2.
```



Como posibilidad alternativa está la de mostrar una cabeza normal de nota en la altura de la nota que debe sonar, con un pequeño círculo que indica que se debe tocar como armónico:

```
d2^\flageolet d_\flageolet
```
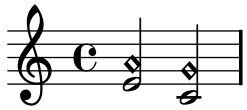


Se puede hacer un círculo de menor tamaño, véase la lista de fragmentos de código en [Referencias para cuerdas sin trastes], página 220.

Armónicos artificiales

La notación de los armónicos artificiales se realiza con dos notas, una con una cabeza normal que indica la posición donde se pisa, y otra con una cabeza en forma de rombo hueco para indicar la posición donde se roza la cuerda (sin pisar) para producir el armónico.

```
<e a\harmonic>2  <c g'\harmonic>
```



Véase también

Glosario musical: Sección “armónicos” in *Glosario Musical*.

Referencia de la notación: [Cabezas de nota especiales], página 27, [Referencias para cuerdas sin trastes], página 220.

Snap (Bartók) pizzicato

Fragmentos de código seleccionados

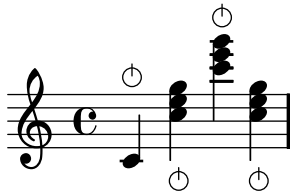
Marca de pizzicato de chasquido (‘pizzicato de Bartók’)

El pizzicato de chasquido (también llamado ‘Pizzicato de Bartók’) es un ‘pizzicato fuerte en que la cuerda se pulsa verticalmente produciendo un chasquido y rebotando en el diapasón del instrumento’ (Wikipedia). Se denota mediante una circunferencia con una línea vertical corta que parte del centro de aquélla hacia fuera. Aunque Lilypond no tiene ninguna instrucción predefinida para crear esta marca, es fácil hacer la definición y colocarla directamente en el archivo de lilypond.

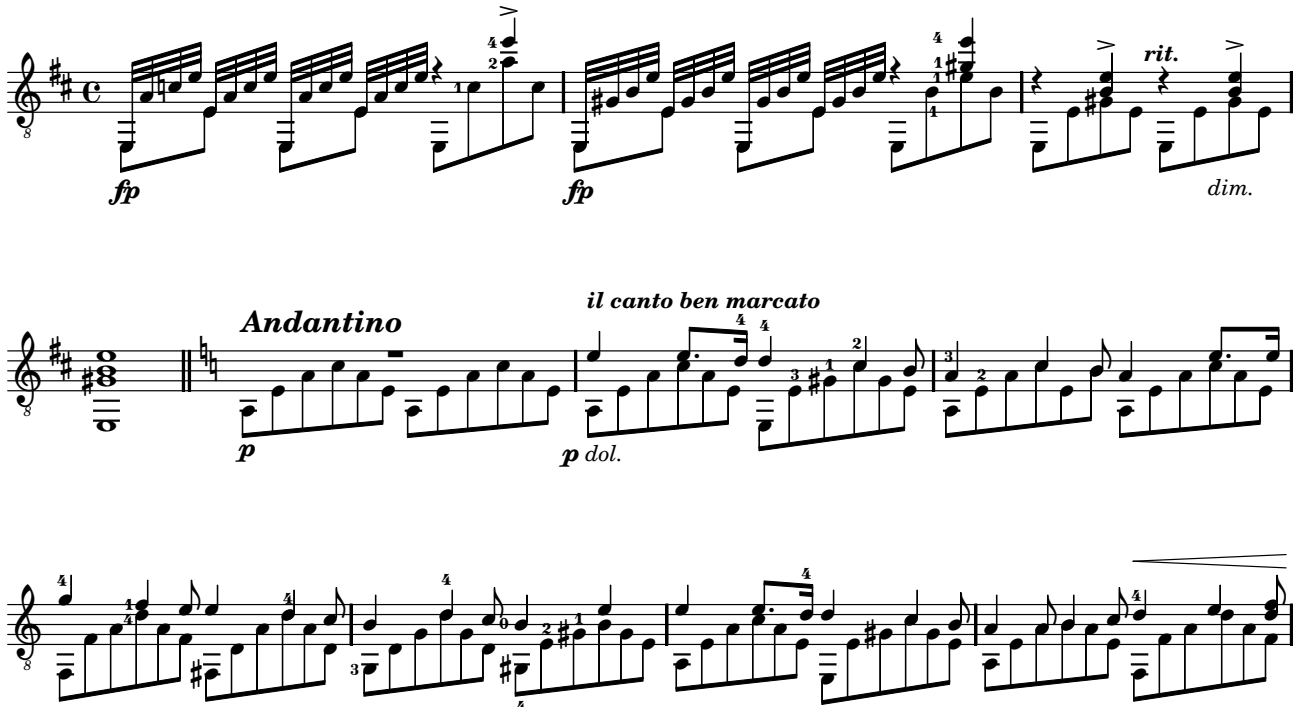
```
#(define-markup-command (snappizz layout props) ()
  (interpret-markup layout props
    (markup #:stencil
      (ly:stencil-translate-axis
        (ly:stencil-add
          (make-circle-stencil 0.7 0.1 #f)
          (ly:make-stencil
            (list 'draw-line 0.1 0 0.1 0 1)
            '(-0.1 . 0.1) '(0.1 . 1)))
        0.7 X))))

pizzicatoSnap = \markup \snappizz

% ahora se puede usar como pizzicatoSnap después de la nota o acorde
% Observe que se requiere una dirección (-, ^ ó _).
\relative c' {
  c4^\pizzicatoSnap
  % Esto NO funciona:
  %<c e g>\pizzicatoSnap
  <c' e g>-\pizzicatoSnap
  <c' e g>^\pizzicatoSnap
  <c, e g>_\pizzicatoSnap
}
```



2.4 Instrumentos de cuerda con trastes



Esta sección trata varios aspectos de la notación musical que son exclusivos de los instrumentos de cuerda con trastes.

2.4.1 Notación común para cuerdas con trastes

Esta sección se ocupa de la notación común exclusiva de los instrumentos de cuerda con trastes.

Referencias para cuerdas con trastes

La notación musical para instrumentos de cuerda con trastes se realiza normalmente en una sola pauta, ya sea en notación musical tradicional o en tablatura. A veces se combinan los dos tipos, y es común en música popular utilizar diagramas de acordes por encima de un pentagrama de notación tradicional. La guitarra y el banjo son instrumentos transpositores, que suenan una octava por debajo de lo escrito. Las partituras para estos instrumentos deben usar la clave de Sol octava baja `"treble_8"`. Otros elementos relativos a los instrumentos de cuerda con trastes se estudian en otras partes de manual:

- Las digitaciones se indican con [Indicaciones de digitación], página 155.
- Las instrucciones para ligaduras *Laissez vibrer* así como ligaduras de unión sobre arpegios y trémolos se describe en [Ligaduras de unión], página 36.
- Las instrucciones para el manejo de varias voces se describe en [Resolución de las colisiones], página 116.

Véase también

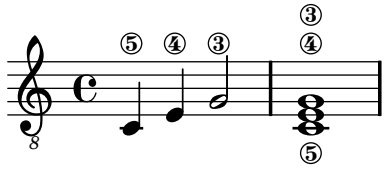
Referencia de la notación: [Indicaciones de digitación], página 155, [Ligaduras de unión], página 36, [Resolución de las colisiones], página 116, [Nombres de instrumentos], página 145, [Escribir música en paralelo], página 123, [Arpegio], página 97, Sección B.10 [Lista de articulaciones], página 513, [Clave], página 12.

Indicación de los números de cuerda

Se puede indicar la cuerda en que se debe tocar una nota añadiendo \ *número* a una nota dentro de una construcción de acorde (<>).

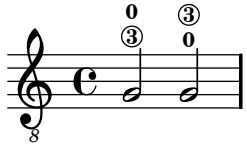
Nota: Los números de cuerda **se deben** definir dentro de una construcción de acorde con ángulos simples, incluso si se trata de una sola nota.

```
\clef "treble_8"
<c\5>4 <e\4> <g\3>2
<c,\5 e\4 g\3>1
```



Cuando se usan al mismo tiempo digitaciones e indicaciones del número de cuerda, su colocación se controla por el orden en que aparecen estos elementos en el código:

```
\clef "treble_8"
<g\3-0>2
<g-0\3>
```

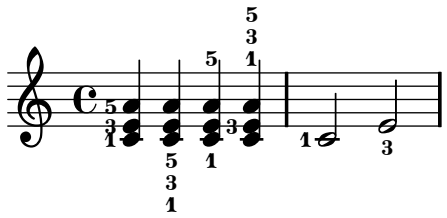


Fragmentos de código seleccionados

Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación.

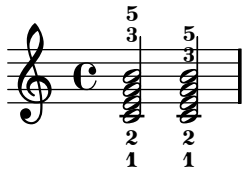
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación se imprimen de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \once \override Fingering #'staff-padding = #'()
  <c-1 e-2 g-3 b-5>2
}
```



Véase también

Referencia de la notación: [Indicaciones de digitación], página 155.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

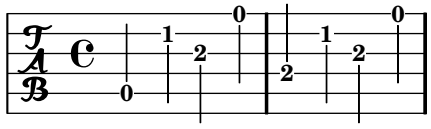
Referencia de funcionamiento interno: Sección “StringNumber” in *Referencia de Funcionamiento Interno*, Sección “Fingering” in *Referencia de Funcionamiento Interno*.

Tablaturas predeterminadas

La notación de tablatura se utiliza para la notación de música de instrumentos de cuerda pulsada. Las alturas no se denotan mediante cabezas de nota, sino mediante números que indican sobre qué cuerda y traste se debe tocar la nota. LilyPond contempla las tablaturas de forma limitada.

El número de cuerda asociado a una nota se proporciona en la forma de una barra invertida seguida por un número, p.ej. `c4\3` es un Do negra sobre la tercera cuerda. Por defecto, la cuerda 1 es la más aguda, y la afinación normal es la afinación estándar de guitarra (con 6 cuerdas). Las notas se imprimen como tablatura, usando lo contextos Sección “TabStaff” in *Referencia de Funcionamiento Interno* y Sección “TabVoice” in *Referencia de Funcionamiento Interno*:

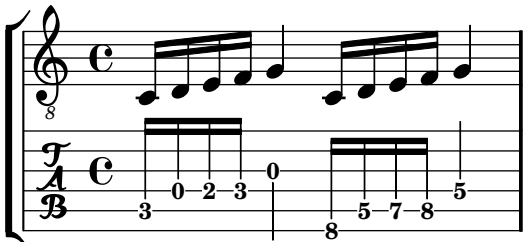
```
\new TabStaff {
  a,4\5 c'\2 a\3 e'\1
  e\4 c'\2 a\3 e'\1
}
```



Si se especifica ninguna cuerda para una nota, se le asigna la cuerda más aguda que produce la nota con un número de traste mayor o igual que el valor de `minimumFret` (traste mínimo). El valor predeterminado de `minimumFret` es cero.

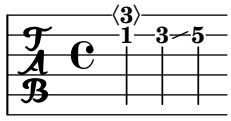
```
\new StaffGroup <<
   \new Staff \relative c {
     \clef "treble_8"
```

```
    c16 d e f g4
    c,16 d e f g4
  }
  \new TabStaff \relative c {
    c16 d e f g4
    \set TabStaff.minimumFret = #5
    c,16 d e f g4
  }
>>
```



Se pueden añadir indicaciones de armónicos y *slides* a la notación de tablatura.

```
\new TabStaff {
  \new TabVoice {
    <c g'\harmonic> d\2\glissando e\2
  }
}
```

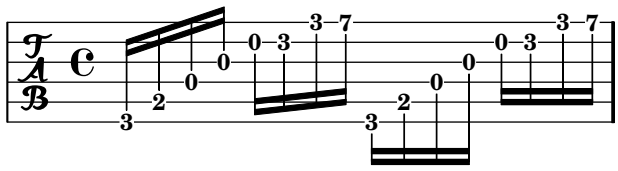


Fragmentos de código seleccionados

Comportamiento de las plicas y las barras de corchea en tablaturas

La dirección de las plicas se contola de la misma forma en la tablatura que en la notación tradicional. Las barras se pueden poner horizontales, como se muestra en este ejemplo.

```
\new TabStaff {
  \relative c {
    g16 b d g b d g b
    \stemDown
    \override Beam #'damping = #+inf.0
    g,,16 b d g b d g b
  }
}
```



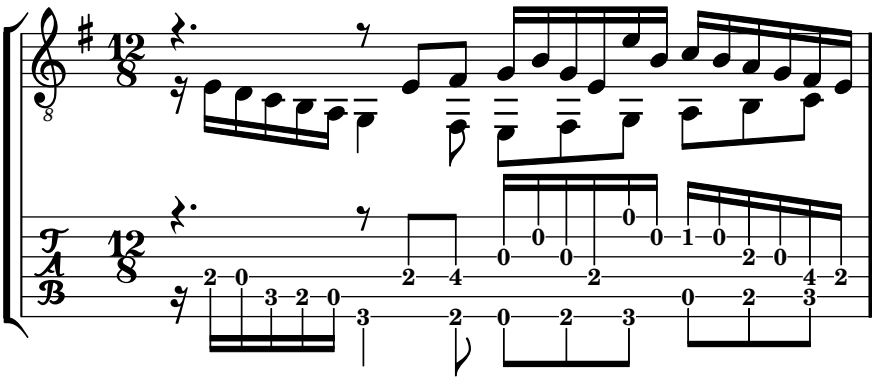
Polifonía en tablaturas

La polifonía se crea de la misma forma en un `TabStaff` que en una pauta normal.

```
superior = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

inferior = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  <<
    \new StaffGroup = "tab with traditional" <<
      \new Staff = "guitarra traditional" <<
        \clef "treble_8"
        \context Voice = "superior" \superior
        \context Voice = "inferior" \inferior
      >>
      \new TabStaff = "guitarra tab" <<
        \context TabVoice = "superior" \superior
        \context TabVoice = "inferior" \inferior
      >>
    >>
  >>
}
```



Véase también

Referencia de la notación: [Plicas], página 160.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

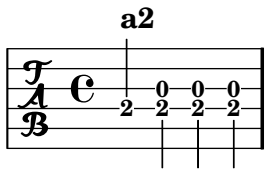
Referencia de funcionamiento interno: Sección “TabNoteHead” in *Referencia de Funcionamiento Interno*, Sección “TabStaff” in *Referencia de Funcionamiento Interno*, Sección “TabVoice” in *Referencia de Funcionamiento Interno*, Sección “Beam” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los acordes no se tratan de una forma especial, y de aquí que el selector automático de la cuerda puede elegir fácilmente la misma cuerda para dos notas del acorde.

Para manejar `\partcombine`, es necesario que `TabStaff` utilice voces expecialmente creadas:

```
melodia = \partcombine { e4 g g g }{ e4 e e e }
<<
  \new TabStaff <<
    \new TabVoice = "uno" s1
    \new TabVoice = "dos" s1
    \new TabVoice = "comun" s1
    \new TabVoice = "solo" s1
    { \melodia }
  >>
>>
```



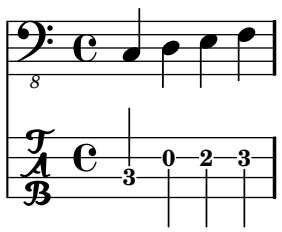
Los efectos especiales de guitarra se limitan a armónicos y *slides*.

Tablaturas personalizadas

La tablatura en LilyPond calcula automáticamente el traste para cada nota, basándose en la cuerda a que está asignada la nota. Para hacerlo, es necesario especificar la afinación de las cuerdas. La afinación de las cuerdas se da en la propiedad `StringTunings`.

LilyPond trae afinaciones predefinidas para el banjo, la mandolina, la guitarra y el bajo. Lilypond establece automáticamente la transposición correcta para las afinaciones predefinidas. El ejemplo siguiente es para bajo, que suena una octaba por debajo de lo escrito.

```
<<
  \new Staff {
    \clef "bass_8"
    \relative c, {
      c4 d e f
    }
  }
  \new TabStaff {
    \set TabStaff.stringTunings = #bass-tuning
    \relative c, {
      c4 d e f
    }
  }
>>
```



La afinación por omisión es `guitar-tuning` (la afinación estándar Mi-La-Re-Sol-Si-Mi). Otras afinaciones predefinidas son `guitar-open-g-tuning` (Sol Mayor al aire, Re-Sol-Re-Sol-Si-Re), `mandolin-tuning` (mandolina) y `banjo-open-g-tuning` (banjo con Sol Mayor al aire). The predefined string tunings are found in `scm/output-lib.scm`.

Una afinación de las cuerdas es una lista de Scheme de alturas, una por cada cuerda, ordenadas de 1 a N, donde la cuerda 1 está arriba en la pauta de tablatura y la cuerda N está abajo. Esto resulta generalmente en un orden de la nota más aguda a la más grave, pero algunos instrumentos (p.ej. el ukelele) no tienen las cuerdas dispuestas en orden de altura.

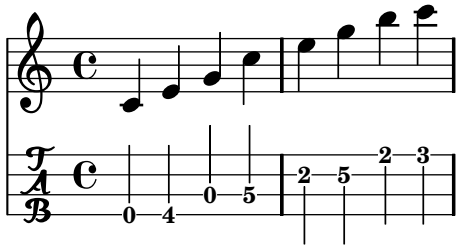
La altura de una cuerda en una lista de afinación de cuerdas es el intervalo que forma la cuerda al aire con el Do central, medido en semitonos. La altura debe ser un número entero. Lilypond calcula la altura real de la cuerda añadiendo la altura de la afinación a la altura real del Do central.

LilyPond calcula automáticamente el número de cuerdas de la pauta `TabStaff` como el número de elementos de la lista `stringTunings`.

Se puede crear cualquier afinación de cuerdas deseada. Por ejemplo, podemos definir una afinación para un instrumento de cuatro cuerdas con las alturas `a''`, `d''`, `g'` y `c'`:

```
misNotas = {
    c'4 e' g' c'' |
    e'' g'' b'' c'''
}

<<
  \new Staff {
    \clef treble
    \misNotas
  }
  \new TabStaff {
    \set TabStaff.stringTunings = #'(21 14 7 0)
    \misNotas
  }
>>
```



Véase también

Archivos de inicio: ‘`scm/output-lib.scm`’.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Tab_note_heads_engraver” in *Referencia de Funcionamiento Interno*.

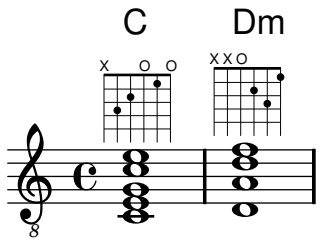
Marcas de diagramas de trastes

Se pueden añadir diagramas de trastes a la música como elementos de marcado sobre la nota deseada. El marcado contiene información sobre el diagrama de trastes deseado. Existen tres

interfaces distintos de marcado de diagramas de trastes: standard (estándar), terse (escueto) y verbose (prolijo). Los tres interfaces producen marcados equivalentes, pero tienen cantidades variables de información en la cadena de marcado. Hay más detalles sobre los interfaces de marcado en Sección B.8 [Text markup commands], página 476.

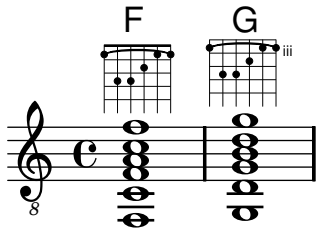
La cadena de marcado de los diagramas estándar de trastes indica el número de la cuerda y el número de traste en que se coloca cada uno de los puntos sobre la cuerda. Además se pueden indicar cuerdas al aire y cuerdas mudas (que no se tocan).

```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram #"6-x;5-3;4-2;3-o;2-1;1-o;"
    < d a d' f'> ^\markup
      \fret-diagram #"6-x;5-x;4-o;3-2;2-3;1-1;"
  }
>>
```



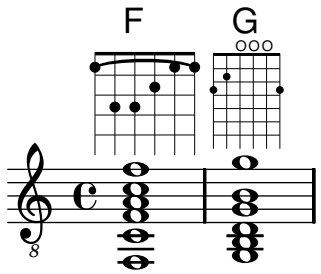
Las indicaciones de cejilla se pueden añadir al diagrama a partir de la cadena de marcado del diagrama de trastes.

```
<<
  \context ChordNames {
     \chordmode {
       f1 g
     }
  }
  \context Staff {
    \clef "treble_8"
    < f, c f a c' f'>1 ^\markup
      \fret-diagram #"c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    < g, d g b d' g'> ^\markup
      \fret-diagram #"c:6-1-3;6-3;5-5;4-5;3-4;2-3;1-3;"
  }
>>
```



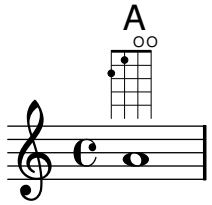
El tamaño del diagrama de trastes y el número de trastes del diagrama, se puede cambiar en la cadena de marcado `fret-diagram`.

```
<<
  \context ChordNames {
     \chordmode {
       f1 g
     }
  }
  \context Staff {
    \clef "treble_8"
    < f, c f a c' f'>1 ^\markup
      \fret-diagram #"s:1.5;c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    < g, b, d g b g'> ^\markup
      \fret-diagram #"h:6;6-3;5-2;4-o;3-o;2-o;1-3;"
  }
>>
```



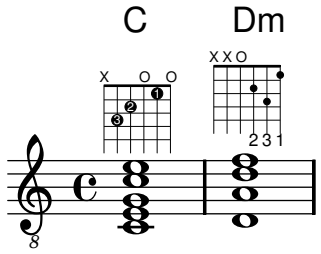
Se puede cambiar el número de cuerdas de un diagrama de trastes para que se adapte a distintos instrumentos como el bajo y el ukelele, con la cadena de marcado del diagrama.

```
<<
  \context ChordNames {
     \chordmode {
       a1
     }
  }
  \context Staff {
        %% Un acorde de ukelele
    a'1 ^\markup \fret-diagram #"w:4;4-2-2;3-1-1;2-o;1-o;"
  }
>>
```



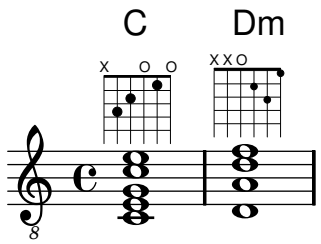
Se pueden apadir indicaciones de digitación, y la posición de las etiquetas de los dedos se puede controlar mediante la cadena de marcado del diagrama.

```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram #"f:1;6-x;5-3-3;4-2-2;3-o;2-1-1;1-o;"
    < d a d' f'> ^\markup
      \fret-diagram #"f:2;6-x;5-x;4-o;3-2-2;2-3-3;1-1-1;"
  }
>>
```



Se puede controlar el radio y la posición de los puntos con la cadena de marcado `fret-diagram`.

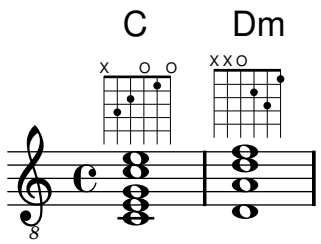
```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram #"d:0.35;6-x;5-3;4-2;3-o;2-1;1-o;"
    < d a d' f'> ^\markup
      \fret-diagram #"p:0.2;6-x;5-x;4-o;3-2;2-3;1-1;"
  }
>>
```



La cadena de marcado de diagrama de trastes escueta, `fret-diagram-terse`, omite los números de la cuerda; el número de cuerda viene implícito por la presencia del punto y coma. Hay un punto y coma por cada cuerda del diagrama. El primer punto y coma corresponde

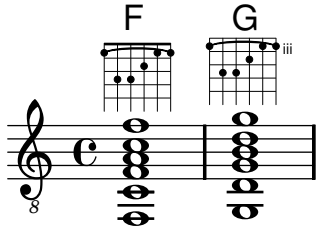
al número de cuerda más alto, y el último punto y coma corresponde a la primera cuerda. Se pueden indicad cuerdas mudas, al aire y números de traste.

```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram-terse #"x;3;2;o;1;o;"
    < d a d' f'> ^\markup
      \fret-diagram-terse #"x;x;o;2;3;1;"
  }
>>
```



Se pueden incluir números de cejilla en la cadena de marcado escueta `fret-diagram-terse`.

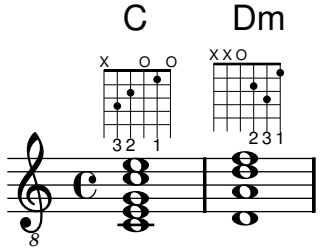
```
<<
  \context ChordNames {
     \chordmode {
       f1 g
     }
  }
  \context Staff {
    \clef "treble_8"
    < f, c f a c' f'>1 ^\markup
      \fret-diagram-terse #"1-(;3;3;2;1;1-);"
    < g, d g b d' g'> ^\markup
      \fret-diagram-terse #"3-(;5;5;4;3;3-);"
  }
>>
```



Se pueden incluir indicaciones de digitación en la cadena de marcado escueta `fret-diagram-terse`

```
<<
```

```
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \override Voice.TextScript
      #'(fret-diagram-details finger-code) = #'below-string
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram-terse #"x;3-3;2-2;o;1-1;o;"
    < d a d' f'> ^\markup
      \fret-diagram-terse #"x;x;o;2-2;3-3;1-1;"
  }
>>
```

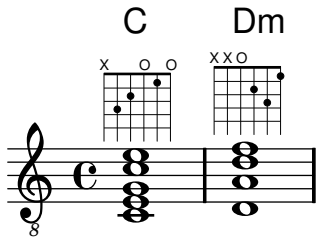


Otras propiedades de los diagramas de trastes se deben ajustar usando `\override` al utilizar el marcado escueto con fret-diagram-terse.

La cadena de marcado prolija `fret-diagram-verbose` está en el formato de una lista de Scheme. Cada elemento de la lista indica la colocación un elemento sobre el diagrama de trastes.

```
<<  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram-verbose #'(
        (mute 6)
        (place-fret 5 3)
        (place-fret 4 2)
        (open 3)
        (place-fret 2 1)
        (open 1)
      )
    < d a d' f'> ^\markup
      \fret-diagram-verbose #'(
        (mute 6)
        (mute 5)
        (open 4)
        (place-fret 3 2)
        (place-fret 2 3)
        (place-fret 1 1)
```

```
      )
  }
>>
```

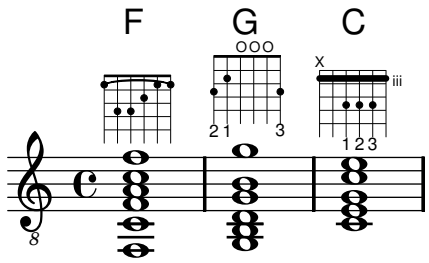


Se pueden incluir digitaciones y cejillas en una cadena de marcado prolija `fret-diagram-verbose`. Es exclusiva del interfaz de fret-diagram-verbose la indicación 'capo' que se puede colocar sobre el diagrama de posición. La indicación de capo es una barra gruesa que cubre todas las cuerdas. El traste que tiene el capo es el más bajo del diagrama de posición.

```
<<
  \context ChordNames {
     \chordmode {
       f1 g c
     }
  }
  \context Staff {
    \clef "treble_8"
    \override Voice.TextScript
      #'(fret-diagram-details finger-code) = #'below-string

    < f, c f a c' f'>1 ^\markup
      \fret-diagram-verbose #'(
        (place-fret 6 1)
        (place-fret 5 3)
        (place-fret 4 3)
        (place-fret 3 2)
        (place-fret 2 1)
        (place-fret 1 1)
        (barre 6 1 1)
      )
    < g, b, d g b g'> ^\markup
      \fret-diagram-verbose #'(
        (place-fret 6 3 2)
        (place-fret 5 2 1)
        (open 4)
        (open 3)
        (open 2)
        (place-fret 1 3 3)
      )
    < c e g c' e'> ^\markup
      \fret-diagram-verbose #'(
        (capo 3)
        (mute 6)
        (place-fret 4 5 1)
        (place-fret 3 5 2)
```

```
        (place-fret 2 5 3)
      )
   }
>>
```



Todas las otras propiedades de diagrama de trastes se deben ajustar utilizando `\override` cuando se usa el marcado prolijo con fret-diagram-verbose.

El aspecto gráfico de un diagrama de trastes se puede personalizar según las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*. Para un marcado de diagrama de trastes, las propiedades del interface pertenecen a `Voice.TextScript`.

Fragmentos de código seleccionados

Personalizar diagramas de posiciones de marcado

Se pueden establecer las propiedades de los diagramas de posiciones a través de `'fret-diagram-details`. Para los diagramas de posiciones de marcado, se pueden aplicar overrides (sobreescrituras) al objeto `Voice.TextScript` o directamente al elemento de marcado.

```
<<
  \chords { c1 c c d }

  \new Voice = "melo" {
    \textLengthOn
    % Fijar propiedades globales del diagrama de posición
    \override TextScript #'size = #'1.2
    \override TextScript
      #'(fret-diagram-details finger-code) = #'in-dot
    \override TextScript
      #'(fret-diagram-details dot-color) = #'white

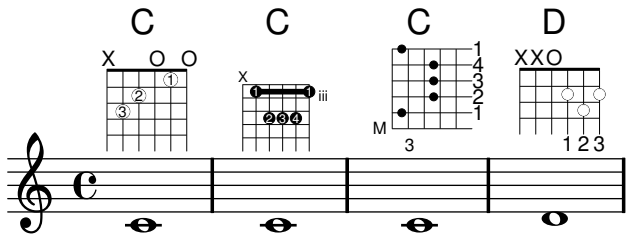
    %% Do mayor para guitarra, sin cejilla, usar predeterminados
       % estilo escueto
    c'1^\markup { \fret-diagram-terse #"x;3-3;2-2;o;1-1;o;" }

    %% Do mayor para guitarra, cejilla en el tercer traste
       % estilo prolijo
       % tamaño 1.0
       % leyenda de posición en romana, leyendas de dedos debajo de las cuerdas, cejilla re
    c'1^\markup {
      % tamaño estándar
      \override #'(size . 1.0) {
        \override #'(fret-diagram-details . (
                     (number-type . roman-lower)
                     (finger-code . in-dot)
```

```
                    (barre-type . straight))) {
          \fret-diagram-verbose #'((mute 6)
                                   (place-fret 5 3 1)
                                   (place-fret 4 5 2)
                                   (place-fret 3 5 3)
                                   (place-fret 2 5 4)
                                   (place-fret 1 3 1)
                                   (barre 5 1 3))
        }
      }
    }

    %% Do mayor para guitarra, cejilla en el tercer traste
       % estilo prolijo
       % orientación apaisada, números arábigos, M para cuerda muda
       % sin cejilla, leyenda de posición abajo o izquierda, fuente pequeña de indicación de
    c'1^\markup {
      \override #'(fret-diagram-details . (
                   (finger-code . below-string)
                   (number-type . arabic)
                   (label-dir . -1)
                   (mute-string . "M")
                   (orientation . landscape)
                   (barre-type . none)
                   (xo-font-magnification . 0.4)
                   (xo-padding . 0.3))) {
        \fret-diagram-verbose #'((mute 6)
                                 (place-fret 5 3 1)
                                 (place-fret 4 5 2)
                                 (place-fret 3 5 3)
                                 (place-fret 2 5 4)
                                 (place-fret 1 3 1)
                                 (barre 5 1 3))
      }
    }

    %% acorde de Re sencillo
       % estilo escueto
       % puntos mayores, centrados, menos trastes
       % leyenda debajo de la cuerda
    d'1^\markup {
      \override #'(fret-diagram-details . (
                   (finger-code . below-string)
                   (dot-radius . 0.35)
                   (dot-position . 0.5)
                   (fret-count . 3))) {
        \fret-diagram-terse #"x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```



Véase también

Referencia de la notación: Sección B.8 [Text markup commands], página 476.

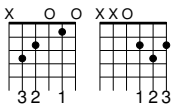
Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*.

Diagramas predefinidos de trastes

Se pueden imprimir los diagramas de trastes usando el contexto `FretBoards`. De forma predeterminada, el contexto `FretBoards` imprime diagramas de trastes que están almacenados en una tabla de búsqueda:

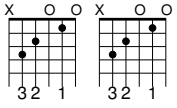
```
\include "predefined-guitar-fretboards.ly"
\context FretBoards {
  \chordmode {
    c1 d
  }
}
```



Los diagramas de traste predefinidos están en el archivo `predefined-guitar-fretboards.ly`. Los diagramas de trastes se almacenan tomando como base las notas de un acorde y el valor de `stringTunings` que se está utilizando en cada momento. `predefined-guitar-fretboards.ly` contiene diagramas de traste predefinidos sólo para `guitar-tuning`. Los diagramas de traste predefinidos se pueden añadir para otros instrumentos u otras afinaciones siguiendo los ejemplos que aparecen en `predefined-guitar-fretboards.ly`.

Las notas de los acordes se pueden introducir como música simultánea o bein usando el modo de acordes (véase [Panorámica del modo de acordes], página 264).

```
\include "predefined-guitar-fretboards.ly"
\context FretBoards {
  \chordmode {c1}
  <c' e' g'>1
}
```

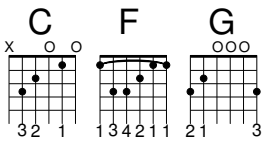


Es corriente que se impriman juntos los nombres de acorde y los diagramas de traste. Esto se puede hacer poniendo un contexto de nombres de acorde `ChordNames` en paralelo con un contexto de trastes `FretBoards` y dando a los dos contextos el mismo contenido musical.

```
\include "predefined-guitar-fretboards.ly"
misAcordes = \chordmode{
  c1 f g
```

```
}

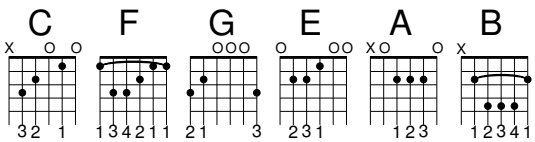
<<
  \context ChordNames {
    \misAcordes
  }
  \context FretBoards {
    \misAcordes
  }
>>
```



Los diagramas de traste predefinidos se pueden transportar, en la medida en que esté almacenado un diagrama para el acorde transportado, en la tabla de diagramas de traste.

```
\include "predefined-guitar-fretboards.ly"
misAcordes = \chordmode{
  c1 f g
}

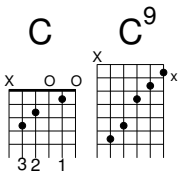
miListaDeAcordes = {
  \misAcordes
  \transpose c e { \misAcordes}
}
<<
  \context ChordNames {
    \miListaDeAcordes
  }
  \context FretBoards {
    \miListaDeAcordes
  }
>>
```



La tabla de diagramas de traste predefinidos contiene siete acordes (mayor, menor, aumentado, disminuido, séptima dominante, séptima mayor y menor séptima) para 17 tonalidades distintas. Se puede ver una lista completa de los diagramas de traste predefinidos en Sección B.3 [Diagramas predefinidos de trastes], página 456. Si no hay una entrada en la tabla para un acorde, el grabador FretBoards calcula un cadena de diagrama fret-diagram usando la funcionalidad de diagramas automáticos que se describe en [Diagramas de traste automáticos], página 245.

```
\include "predefined-guitar-fretboards.ly"
misAcordes = \chordmode{
  c1 c:9
}
```

```
<<
  \context ChordNames {
    \misAcordes
  }
  \context FretBoards {
    \misAcordes
  }
>>
```



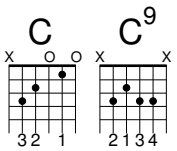
Se pueden añadir diagramas de posiciones a la tabla de diagramas de posiciones. Para añadir un diagrama debemos especificar el acorde del diagrama, la afinación utilizada y una definición del diagrama. La definición del diagrama puede ser una cadena de definición escueta fret-diagram-terse o una lista de marcados prolija fret-diagram-verbose.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram \chordmode {c:9}
                        #guitar-tuning
                        #"x;3-2;2-1;3-3;3-4;x;"

misAcordes = \chordmode{
  c1 c:9
}

<<
  \context ChordNames {
    \misAcordes
  }
  \context FretBoards {
    \misAcordes
  }
>>
```



Se pueden almacenar distintos diagramas de traste para el mismo acorde usando distinas octavas para las notas.

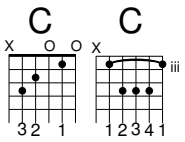
```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram \chordmode {c'}
                        #guitar-tuning
                        #(offset-fret 2 (chord-shape 'bes guitar-tuning))

misAcordes = \chordmode{
```

```
  c1 c'
}

<<
  \context ChordNames {
    \misAcordes
  }
  \context FretBoards {
    \misAcordes
  }
>>
```



Además de los diagramas de traste, LilyPond almacena una lista interna de formas de acorde. Las formas de acorde son diagramas de traste que se pueden desplazar por el mástil para dar acordes distintos. Se pueden añadir formas de acorde a la lista interna y luego usarlas para definir diagramas de posición predefinidos. Dado que se pueden mover a distintas posiciones dentro del mástil, las formas de acorde normalmente no contienen cuerdas al aire. Como los diagramas de posiciones, las formas de acorde se pueden introducir como cadenas escuetas fret-diagram-terse o como listas de marcado prolijas fret-diagram-verbose.

```
\include "predefined-guitar-fretboards.ly"

% añadir una forma de acorde nueva

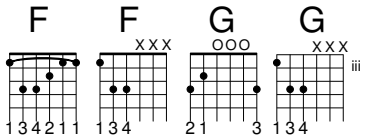
\addChordShape #'powerf #guitar-tuning #"1-1;3-3;3-4;x;x;x;"

% añadir acordes nuevos basados en la forma de acorde principal

\storePredefinedDiagram \chordmode {f'}
                        #guitar-tuning
                        #(chord-shape 'powerf guitar-tuning)
\storePredefinedDiagram \chordmode {g'}
                        #guitar-tuning
                        #(offset-fret 2 (chord-shape 'powerf guitar-tuning))

misAcordes = \chordmode{
  f1 f' g g'
}

<<
  \context ChordNames {
    \misAcordes
  }
  \context FretBoards {
    \misAcordes
  }
>>
```



El aspecto gráfico de un diagrama de traste se puede personalizar según las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*. Para un diagrama de trastes predefinido, las propiedades de interface pertenecen a `FretBoards.FretBoard`.

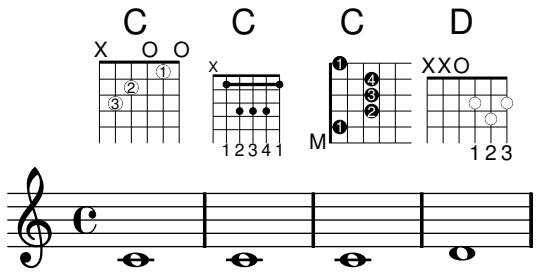
Fragmentos de código seleccionados

Personalizar los diagramas de posiciones

Se pueden establecer las propiedades de los diagramas de posiciones de acordes por medio de `'fret-diagram-details`. Para los diagramas de posiciones de FretBoard, se aplican los overrides (sobreescrituras) al objeto `FretBoards.FretBoard`. Como `Voice`, `FretBoards` es un contexto del nivel inferior, y por tanto se puede omitir su nombre en la sobreescritura de propiedades.

```
\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram \chordmode { c' }
                        #guitar-tuning
                        #"x;1-1-(;3-2;3-3;3-4;1-1-);"
<<
  \new ChordNames {
    \chordmode { c1 c c d }
  }
  \new FretBoards {
    % Fijar propiedades globales del diagrama de posición
    \override FretBoards.FretBoard #'size = #'1.2
    \override FretBoard
      #'(fret-diagram-details finger-code) = #'in-dot
    \override FretBoard
      #'(fret-diagram-details dot-color) = #'white
    \chordmode {
      c
      \once \override FretBoard #'size = #'1.0
      \once \override FretBoard
        #'(fret-diagram-details barre-type) = #'straight
      \once \override FretBoard
        #'(fret-diagram-details dot-color) = #'black
      \once \override FretBoard
        #'(fret-diagram-details finger-code) = #'below-string
      c'
      \once \override FretBoard
        #'(fret-diagram-details barre-type) = #'none
      \once \override FretBoard
        #'(fret-diagram-details number-type) = #'arabic
      \once \override FretBoard
        #'(fret-diagram-details orientation) = #'landscape
      \once \override FretBoard
        #'(fret-diagram-details mute-string) = #"M"
      \once \override FretBoard
        #'(fret-diagram-details label-dir) = #LEFT
      \once \override FretBoard
        #'(fret-diagram-details dot-color) = #'black
      c'
```

```
      \once \override FretBoard
        #'(fret-diagram-details finger-code) = #'below-string
      \once \override FretBoard
        #'(fret-diagram-details dot-radius) = #0.35
      \once \override FretBoard
        #'(fret-diagram-details dot-position) = #0.5
      \once \override FretBoard
        #'(fret-diagram-details fret-count) = #3
      d
    }
  }
  \new Voice {
    c'1 c' c' d'
  }
>>
```



Definición de posiciones predefinidas para otros instrumentos

Se pueden añadir diagramas de posiciones predefinidas para instrumentos nuevos además de los estándar que se usan para la guitarra. Este archivo muestra cómo se hace, definiendo una afinación nueva y unas cuantas posiciones para el cuatro venezolano.

Este archivo también muestra cómo se pueden incluir las digitaciones en los acordes que se usan como puntos de referencia para la búsqueda de acordes en la tabla, y mostrarse en el diagrama de posiciones y la tablatura `TabStaff`, pero no en la música.

Estas posiciones no se pueden transportar porque contienen información de las cuerdas. Hay planes para corregir esto en un futuro.

```
% añadir FretBoards para el cuatro venezolano
%   Nota: esta sección se puede poner en un archivo aparte
%      posiciones-predefinidas-del-cuatro.ly
%      e incluirse en todas sus composiciones con \include

afinacionCuatro = #'(11 18 14 9)

sextaRe = { <a\4 b\1 d\3 fis\2> }
reMayor = { <a\4 d\1 d\3 fis \2> }
laSeptMayor = { <a\4 cis\1 e\3 g\2> }
reSeptMayor = { <a\4 c\1 d\3 fis\2> }
solMayor = { <b\4 b\1 d\3 g\2> }

\storePredefinedDiagram \sextaRe
                        #afinacionCuatro
                        #"o;o;o;o;"
\storePredefinedDiagram \reMayor
```

```
                        #afinacionCuatro
                        #"o;o;o;3-3;"
\storePredefinedDiagram \laSeptMayor
                        #afinacionCuatro
                        #"o;2-2;1-1;2-3;"
\storePredefinedDiagram \reSeptMayor
                        #afinacionCuatro
                        #"o;o;o;1-1;"
\storePredefinedDiagram \solMayor
                        #afinacionCuatro
                        #"2-2;o;1-1;o;"

% fin del potencial archivo de inclusión /posiciones-predefinidas-del-cuatro.ly


#(set-global-staff-size 16)

nombresPrimeros = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \sextaRe \reMayor \laSeptMayor \reSeptMayor
  \solMayor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \nombresPrimeros
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }

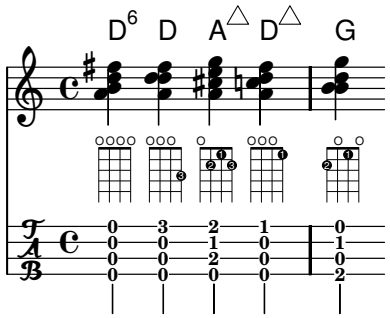
    \new FretBoards {
      \set stringTunings = #afinacionCuatro
      \override FretBoard
        #'(fret-diagram-details string-count) = #'4
      \override FretBoard
        #'(fret-diagram-details finger-code) = #'in-dot
      \primeros
    }

    \new TabStaff \relative c'' {
```

```
      \set TabStaff.stringTunings = #afinacionCuatro
      \primeros
    }

  >>

  \layout {
    \context {
      \Score
      \override SpacingSpanner
        #'base-shortest-duration = #(ly:make-moment 1 16)
    }
  }
  \midi { }
}
```



Véase también

Referencia de la notación: [Tablaturas personalizadas], página 228, [Diagramas de traste automáticos], página 245, [Panorámica del modo de acordes], página 264, Sección B.3 [Diagramas predefinidos de trastes], página 456.

Archivos de inicio: ‘`ly/predefined-guitar-fretboards.ly`’, ‘`ly/predefined-guitar-ninth-fretboards`

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

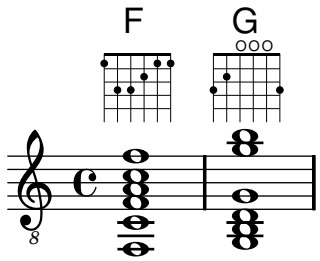
Referencia de funcionamiento interno: Sección “fret-diagram-interface” in *Referencia de Funcionamiento Interno*.

Diagramas de traste automáticos

Se pueden crear diagramas de traste automáticamente a partir de las notas introducidas usando el contexto `FretBoards`. Si no está disponible ningún diagrama predefinido para las notas introducidas en la afinación `stringTunings` activa, este contexto calcula las cuerdas y los trastes que se pueden usar para tocar las notas.

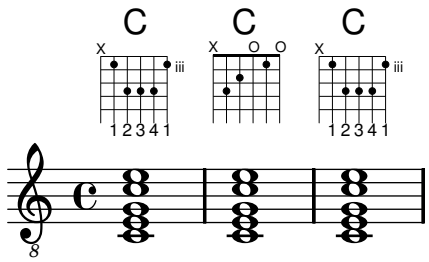
```
<<
  \context ChordNames {
     \chordmode {
       f1 g
     }
  }
  \context FretBoards {
    < f, c f a c' f'>1
    < g,\6 b, d g b g'>
  }
```

```
  \context Staff {
    \clef "treble_8"
    < f, c f a c' f'>1
    < g, b, d g b' g'>
  }
>>
```



Puesto que de forma predeterminada no se carga ningún diagrama predefinido, el comportamiento predeterminado es el cálculo automático de los diagramas de traste. Una vez que los diagramas predeterminados se han cargado, se puede habilitar e inhabilitar el cálculo automático con instrucciones predefinidas:

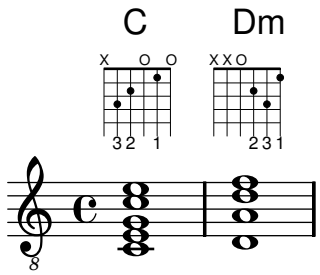
```
\storePredefinedDiagram <c e g c' e'>
                        #guitar-tuning
                        #"x;3-1-(;5-2;5-3;5-4;3-1-1);"
<<
  \context ChordNames {
     \chordmode {
       c1 c c
     }
  }
  \context FretBoards {
    <c e g c' e'>1
    \predefinedFretboardsOff
    <c e g c' e'>
    \predefinedFretboardsOn
    <c e g c' e'>
  }
  \context Staff {
    \clef "treble_8"
    <c e g c' e'>1
    <c e g c' e'>
    <c e g c' e'>
  }
>>
```



A veces el calculador de trastes es incapaz de encontrar un diagrama aceptable. Esto se puede remediar generalmente mediante la asignación manual de una nota a una cuerda. En muchos casos sólo hay que colocar manualmente una nota sobre una cuerda; el resto de las notas se situará en el lugar adecuado por parte del contexto `FretBoards`.

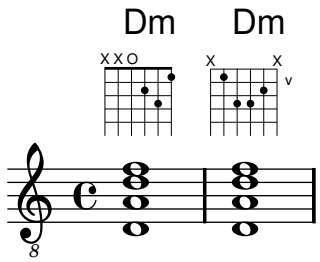
Se pueden añadir digitaciones a los diagramas de traste del contexto FretBoard.

```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context FretBoards {
    < c-3 e-2 g c'-1 e' > 1
    < d a-2 d'-3 f'-1>
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1
    < d a d' f'>
  }
>>
```



El traste mínimo que usar en el cálculo de cuerdas y trastes para el contexto FretBoard se puede fijar con la propiedad `minimumFret`.

```
<<
  \context ChordNames {
     \chordmode {
       d1:m d:m
     }
  }
  \context FretBoards {
    < d a d' f'>
    \set FretBoards.minimumFret = #5
    < d a d' f'>
  }
  \context Staff {
    \clef "treble_8"
    < d a d' f'>
    < d a d' f'>
  }
>>
```



Las cuerdas y los trastes para el contexto `FretBoards` dependen de la propiedad `stringTunings`, que tiene el mismo significado que en el contexto de tablatura TabStaff. Consulte [Tablaturas personalizadas], página 228 para ver más información sobre la propiedad `stringTunings`.

El aspecto gráfico de un diagrama de trastes se puede personalizar de acuerdo a las preferncias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección "fret-diagram-interface" in *Referencia de Funcionamiento Interno*. Para un diagrama de `FretBoards`, las propiedades del interface pertenecen a `FretBoards.FretBoard`.

Instrucciones predefinidas

`\predefinedFretboardsOff`, `\predefinedFretboardsOn`.

Véase también

Referencia de la notación: [Tablaturas personalizadas], página 228.

Fragmentos de código: Sección "Fretted strings" in *Fragmentos de código*.

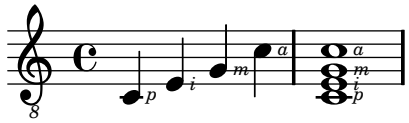
Referencia de funcionamiento interno: Sección "fret-diagram-interface" in *Referencia de Funcionamiento Interno*.

Digitaciones de la mano derecha

Las digitaciones de la mano derecha *p-i-m-a* se deben introducir dentro de una construcción de acorde `<>` para que se impriman en la partitura, incluso si se aplican a una nota suelta.

> **Nota: Debe** haber un guión después de la nota y un espacio antes del ángulo de cierre `>`.

```
\clef "treble_8"
<c-\rightHandFinger #1 >4
<e-\rightHandFinger #2 >
<g-\rightHandFinger #3 >
<c-\rightHandFinger #4 >
<c,-\rightHandFinger #1 e-\rightHandFinger #2
  g-\rightHandFinger #3 c-\rightHandFinger #4 >1
```



Por brevedad, se puede escribir `\rightHandFinger` como algo más corto, como por ejemplo `RH`:

```
#(define RH rightHandFinger)
```

Fragmentos de código seleccionados

Posicionamiento de digitaciones de mano derecha

Es posible ejercer un mayor control sobre la colocación de las digitaciones de la mano derecha estableciendo el valor de una propiedad específica, como se muestra en el ejemplo siguiente.

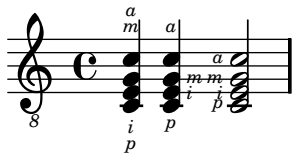
```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"

  \set strokeFingerOrientations = #'(up down)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >4

  \set strokeFingerOrientations = #'(left)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >2
}
```

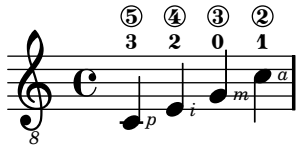


Digitaciones, indicación del número de cuerda y digitaciones de mano derecha

En este ejemplo se combinan las digitaciones de la mano izquierda, indicaciones del número de cuerda y digitaciones de la mano derecha.

```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"
  <c-3\5-\RH #1 >4
  <e-2\4-\RH #2 >4
  <g-0\3-\RH #3 >4
  <c-1\2-\RH #4 >4
}
```



Véase también

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StrokeFinger” in *Referencia de Funcionamiento Interno*.
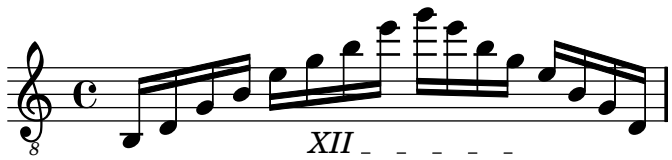
2.4.2 Guitarra

Casi todos los asintos de notación relacionados con la música para guitarra están suficientemente cubiertos en la sección general sobre ajustes de instrumentos de trastes, pero hay algunos más de los que merece la pena hablar aquí. De forma ocasional, los usuarios quieren crear documentos del tipo de cancioneros que tengan sólo la letra de las canciones con indicaciones de acordes sobre ella. Dado que Lilypond es un tipografiador de música, no se recomienda para documentos que no tienen notación musical. Una alternativa mejor es un procesador de texto, editor de texto o, para usuarios con experiencia, un tipografiador como GuitarTeX.

Indicar la posición y la cejilla

Este ejemplo muestra cómo incluir indicaciones de posición y cejilla de guitarra:

```
\clef "treble_8"
b16 d g b e
\textSpannerDown
\override TextSpanner #'(bound-details left text) = #"XII "
  g16\startTextSpan
  b16 e g e b g\stopTextSpan
e16 b g d
```



Véase también

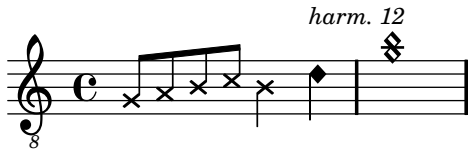
Referencia de la notación: [Extensiones de texto], página 167.

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*, Sección “Expressive marks” in *Fragmentos de código*.

Indicar armónicos y notas tapadas

Se pueden usar cabezas de nota especiales para indicar notas apagadas o armónicos. Los armónicos se suelen explicar de forma más completa con un marcado de texto.

```
\relative c' {
  \clef "treble_8"
  \override Staff.NoteHead #'style = #'cross
  g8 a b c b4
  \override Staff.NoteHead #'style = #'harmonic-mixed
  d^\markup { \italic { \fontsize #-2 { "harm. 12" }}} <g b>1
}
```



Véase también

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

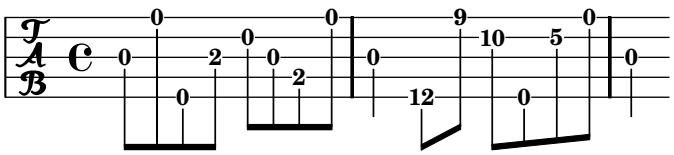
Referencia de la notación: [Cabezas de nota especiales], página 27, Sección B.7 [Estilos de cabezas de nota], página 475.

2.4.3 Banjo

Tablaturas de banjo

LilyPond contempla el banjo de cinco cuerdas de una manera básica. Cuando haga tablaturas para banjo de cinco cuerdas, utilice la función de formato de tablatura de banjo para obtener los números correctos de los trastes para la quinta cuerda:

```
\new TabStaff <<
  \set TabStaff.tablatureFormat = #fret-number-tablature-format-banjo
  \set TabStaff.stringTunings = #banjo-open-g-tuning
  {
    \stemDown
    g8 d' g'\5 a b g e d' |
    g4 d''8\5 b' a'\2 g'\5 e'\2 d' |
    g4
  }
>>
```



En LilyPond se encuentran predefinidas varias afinaciones comunes para el banjo: `banjo-c-tuning` (sol-Do-Sol-Si-Re), `banjo-modal-tuning` (sol-Re-Sol-Do-Re), `banjo-open-d-tuning` (Re Mayor al aire, la-Re-Fa#-La-Re) y `banjo-open-dm-tuning` (Re menor al aire, la-Re-Fa-La-Re).

Estas afinaciones se pueden convertir a afinaciones para bajo de cuatro cuerdas utilizando la función `four-string-banjo`:

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

Véase también

Fragmentos de código: Sección “Fretted strings” in *Fragmentos de código*.

El archivo ‘`scm/output-lib.scm`’ contiene las afinaciones predefinidas para el banjo.

2.5 Percusión

2.5.1 Notación común para percusión

La notación rítmica se emplea primordialmente para la percusión y la batería, pero también se puede utilizar para mostrar los valores rítmicos una melodía.

Referencias para percusión

- La notación de algunas percusiones se puede hacer sobre una pauta de ritmo; esto se estudia en [Mostrar los ritmos de la melodía], página 53 y [Crear instancias de pentagramas nuevos], página 126.
- La salida MIDI se trata en una sección aparte; véase Sección 3.5.6 [Percusión en MIDI], página 343.

Véase también

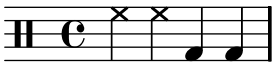
Referencia de la notación: [Mostrar los ritmos de la melodía], página 53, [Crear instancias de pentagramas nuevos], página 126. Sección 3.5.6 [Percusión en MIDI], página 343.

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

Notación básica de percusión

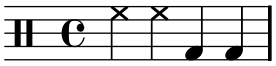
Las notas de percusión se pueden escribir en el modo `\drummode`, que es similar al modo estándar para introducir notas. La manera más fácil de escribir notas de percusión es utilizar la instrucción `\drums`, que crea el contexto y el modo de entrada apropiados para percusión:

```
\drums {
  hihat4 hh bassdrum bd
}
```



Esto es una abreviatura de

```
\new DrumStaff {
  \drummode {
    hihat4 hh bassdrum bd
  }
}
```



Cada elemento de un set de percusión lleva un nombre completo y un nombre abreviado, y los dos se pueden usar en la entrada. La lista completa de nombre de instrumentos de percusión se encuentra en Sección B.11 [Notas de percusión], página 514.

Observe que la notación normal de las notas con altura determinada (como `cis4`) en un contexto `DrumStaff` procuden un mensaje de error. Las claves de percusión se añaden automáticamente al contexto `DrumStaff`, pero también se pueden usar otras claves.

Hay ciertos detalles respecto a cómo está contemplado el MIDI para los instrumentos de percusión; para ver los detalles consulte Sección 3.5.6 [Percusión en MIDI], página 343.

Véase también

Referencia de la notación: Sección 3.5.6 [Percusión en MIDI], página 343, Sección B.11 [Notas de percusión], página 514.
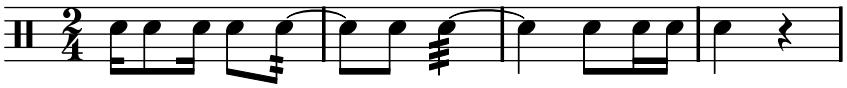
Archivos de inicio: ‘`ly/drumpitch-init.ly`’.

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

Redobles
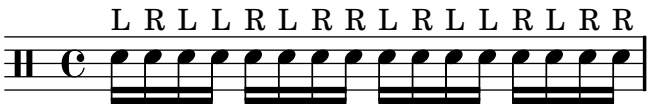
Los redobles se indican mediante tres barras cruzadas en la plica. Para las negras o notas más largas, las tres barras se muestran explícitamente, las corcheas se presentan con dos barras cruzadas (siendo la barra de corchea la tercera), y los redobles más breves que las corcheas tienen una barra cruzada para complementar las barras normales. Esto se consigue mediante la notación de trémolo, `:32`, véase [Repeticiones de trémolo], página 110. He aquí un ejemplo de redobles de caja:

```
\drums {
  \time 2/4
  sn16 sn8 sn16 sn8 sn8:32 ~
  sn8 sn8 sn4:32 ~
  sn4 sn8 sn16 sn16
  sn4 r4
}
```



Los golpes de baqueta se pueden indicar mediante la colocación de `^"R"` o `^"L"` después de la nota. La propiedad `staff-padding` se puede sobreescribir para conseguir una línea de base satisfactoria.

```
\drums {
  \repeat unfold 2 {
    sn16 ^"L" sn^"R" sn^"L" sn^"L" sn^"R" sn^"L"  sn^"R" sn^"R"
  }
}
```



Véase también

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

Percusión afinada

Ciertos instrumentos de percusión de altura determinada (p.ej.: el xilófono, el vibráfono y los timbales) se escriben usando pentagramas normales. Esto se estudia en otras secciones del manual.

Véase también

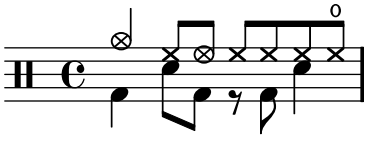
Referencia de la notación: Sección 3.5.6 [Percusión en MIDI], página 343.

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

Pautas de percusión

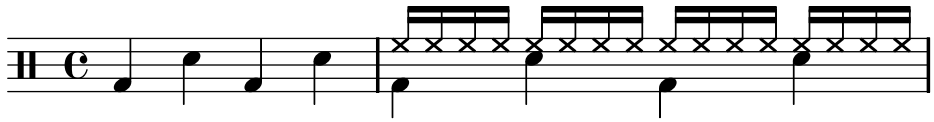
Una parte de percusión para más de un instrumento, normalmente utiliza una pauta de varias líneas donde cada posición dentro de la pauta se refiere a un elemento de percusión. Para tipografiar la música, se deben interpretar las notas dentro de los contextos Sección “DrumStaff” in *Referencia de Funcionamiento Interno* y Sección “DrumVoice” in *Referencia de Funcionamiento Interno*.

```
arriba = \drummode {
  crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat
}
abajo = \drummode {
  bassdrum4 snare8 bd r bd sn4
}
\new DrumStaff <<
  \new DrumVoice { \voiceOne \arriba }
  \new DrumVoice { \voiceTwo \abajo }
>>
```



El ejemplo anterior muestra una notación polifónica prolija. La notación polifónica abreviada, descrita en Sección “I’m hearing Voices” in *Manual de Aprendizaje*, también se puede usar si las voces se instancian primero a mano. Por ejemplo,

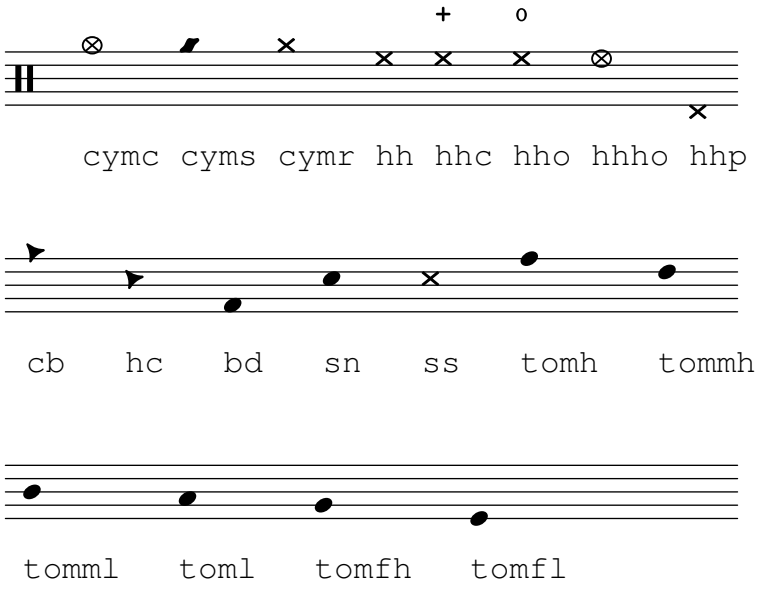
```
\new DrumStaff <<
  \new DrumVoice = "1" { s1*2 }
  \new DrumVoice = "2" { s1*2 }
  \drummode {
    bd4 sn4 bd4 sn4
    << {
      \repeat unfold 16 hh16
    } \\ {
      bd4 sn4 bd4 sn4
    } >>
  }
>>
```



También existen otras posibilidades en lo referente a la disposición. Para usarlas, establezca la propiedad `drumStyleTable` en el contexto Sección "DrumVoice" in *Referencia de Funcionamiento Interno*. Se han predefinido las siguientes variables:

`drums-style`

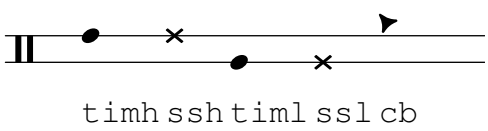
Es la opción por defecto. Tipografía un típico set de percusión sobre pentagrama:



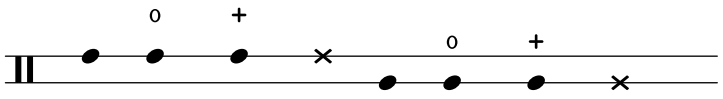
El esquema de percusión contempla seis tambores graves (toms) distintos. Cuando haya menos toms, sencillamente seleccione aquellos que producen el resultado deseado, es decir, para tener toms en las tres líneas centrales utilizará `tommh`, `tomml` y `tomfh`.

`timbales-style`

Esto tipografía timbales en una pauta de dos líneas:

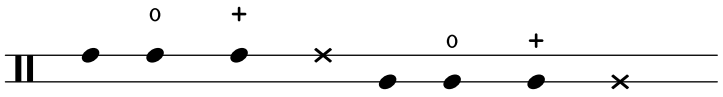


`congas-style`
: Esto tipografía congas en una pauta de dos líneas:



cgh cgho cghm ssh cgl cglo cglm ssl

`bongos-style`
: Esto tipografía bongos sobre una pauta de dos líneas:



boh boho bohm ssh bol bolo bolm ssl

`percussion-style`
: Para tipografiar toda clase de percusiones simples sobre pautas de una línea:



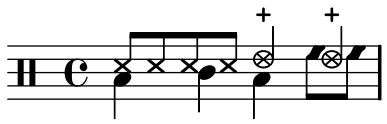
tri trio trim gui guis guil cb cl tamb cab mar hc

Pautas de percusión personalizadas

Si no le gusta ninguna de las listas predefinidas, puede definir su propia lista al principio de su archivo

```
#(define mydrums '(
         (bassdrum     default   #f         -1)
         (snare        default   #f         0)
         (hihat        cross     #f         1)
         (pedalhihat   xcircle   "stopped"  2)
         (lowtom       diamond   #f         3)))
arriba = \drummode { hh8 hh hh hh hhp4 hhp }
abajo = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff <<
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \new DrumVoice { \voiceOne \arriba }
  \new DrumVoice { \voiceTwo \abajo }
>>
```



Fragmentos de código seleccionados

He aquí algunos ejemplos adicionales:

Dos bloques de madera, escritos con wbh (*woodblock-high*, bloque alto) y wbl (*woodblock-low* bloque bajo)

```
% Estas líneas definen la posición de los bloques en la pauta;
% si quiere, puede cambiarlo o usar cabezas especiales
% para los bloques.
#(define mydrums '((hiwoodblock default #t  3)
                   (lowoodblock default #t -2)))

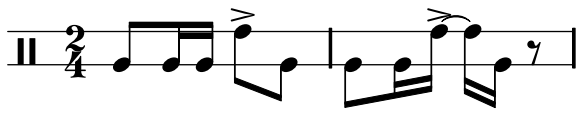
pautaBloques = {
  % Esto define una pauta con sólo dos líneas.
  % También define las posiciones de las dos líneas.
  \override Staff.StaffSymbol #'line-positions = #'(-2 3)

  % Esto es necesario; si no se pone, la línea divisoria sería demasiado corta.
  \override Staff.BarLine #'bar-size = #3
}

\new DrumStaff {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  % con esto cargamos la nueva tabla de estilos de percusión
  \pautaBloques

  \drummode {
    \time 2/4
    wbl8 wbl16 wbl wbh8-> wbl |
    wbl8 wbl16 wbh-> ~ wbh wbl16 r8 |
  }
}
```



Observe que en este caso especial se debe alterar la longitud de la línea divisoria con `\override Staff.BarLine #'bar-size #number`. En caso contrario resulta demasiado corta. También debe definir las posiciones de las dos líneas del pentagrama. Para más información sobre estos delicados asuntos, consulte [El símbolo del pentagrama], página 133.

Una pandereta, escrita mediante 'tamb' (*tambourine*:

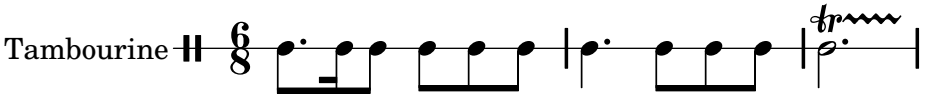
```
#(define mydrums '((tambourine default #t 0)))

pautaPandereta = {
  \override Staff.StaffSymbol #'line-positions = #'( 0 )
  \override Staff.BarLine #'bar-size = #3
  \set DrumStaff.instrumentName = #"Tambourine"
}

\new DrumStaff {
```

```
  \pautaPandereta
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \drummode {
    \time 6/8
    tamb8. tamb16 tamb8 tamb tamb tamb |
    tamb4. tamb8 tamb tamb |
    % el truco con la duración escalada y el silencio más corto
    % es necesario para la correcta finalización del trino.
    tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
  }
}
```



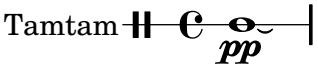
Música para gong, introducida con 'tt' (tam-tam):

```
#(define mydrums '((tamtam default #t 0)))

pautaGong = {
  \override Staff.StaffSymbol #'line-positions = #'( 0 )
  \override Staff.BarLine #'bar-size = #3
  \set DrumStaff.instrumentName = #"Tamtam"
}

\new DrumStaff {
  \pautaGong
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \drummode {
    tt 1 \pp \laissezVibrer
  }
}
```
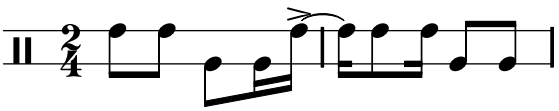


Dos campanas, introducidas con 'cb' (*cowbell*, cencerro) y 'rb' (*ridebell*, campana normal)

```
#(define mydrums '((ridebell default #t  3)
                   (cowbell  default #t -2)))

pautaCampana = {
  \override DrumStaff.StaffSymbol #'line-positions = #'(-2 3)
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.BarLine #'bar-size = #3
  \set DrumStaff.instrumentName = #"Different Bells"
}

\new DrumStaff {
```

```
  \pautaCampana
  \drummode {
    \time 2/4
    rb8 rb cb cb16 rb-> ~ |
    rb16 rb8 rb16 cb8 cb |
  }
}
```

Different Bells 

Aquí un breve ejemplo del maestro Stravinsky (procedende de la 'Historia del soldado')

```
#(define mydrums '((bassdrum   default #t  4)
                   (snare      default #t -4)
                   (tambourine default #t  0)))

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}

percusionA = {
  \context DrumVoice <<
    { \global }
    { \drummode {
        \autoBeamOff
        \stemDown sn8 \stemUp tamb s8 |
        sn4 \stemDown sn4 |
        \stemUp tamb8 \stemDown sn8 \stemUp sn16 \stemDown sn \stemUp sn8 |
        \stemDown sn8 \stemUp tamb s8 |
        \stemUp sn4 s8 \stemUp tamb
      }
    }
  >>
}

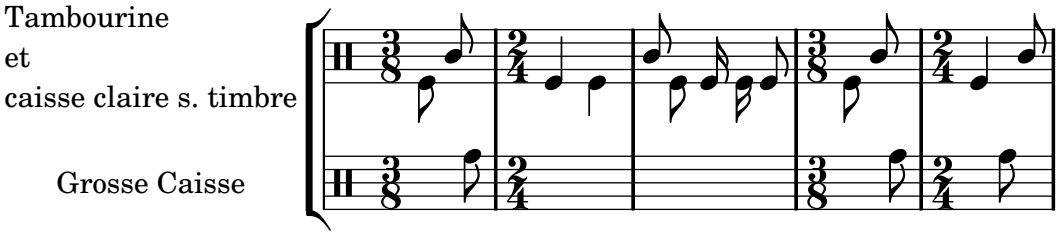
percusionB = {
  \drummode {
    s4 bd8 s2*2 s4 bd8 s4 bd8 s8
  }
}

\layout {
  indent = #40
}

\score {
  \new StaffGroup <<
    \new DrumStaff {
```

```
      \set DrumStaff.instrumentName = \markup {
        \column {
          "Tambourine"
          "et"
          "caisse claire s. timbre"
        }
      }
      \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
      \percusionA
    }

   \new DrumStaff {
     \set DrumStaff.instrumentName = #"Grosse Caisse"
     \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
     \percusionB }
  >>
}
```



Véase también

Fragmentos de código: Sección “Percussion” in *Fragmentos de código*.

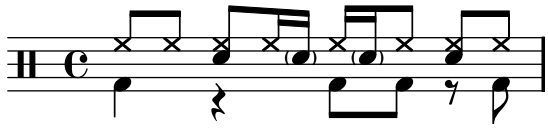
Referencia de funcionamiento interno: Sección “DrumStaff” in *Referencia de Funcionamiento Interno*, Sección “DrumVoice” in *Referencia de Funcionamiento Interno*.

Notas fantasma

Las notas fantasma para la batería e instrumentos de percusión se pueden crear utilizando la instrucción `\parenthesize` detallada en [Paréntesis], página 160. Sin embargo, el modo por defecto `\drummode` no incluye el complemento grabador `Parenthesis_engraver` que permite esto.

```
\new DrumStaff \with {
  \consists "Parenthesis_engraver"
}
<<
  \context DrumVoice  = "1" { s1 }
  \context DrumVoice  = "2" { s1 }
  \drummode {
    <<
      {
        hh8[ hh] <hh sn> hh16
        < \parenthesize sn > hh
        < \parenthesize sn > hh8 <hh sn> hh
      } \\
      {
        bd4 r4 bd8 bd r8 bd
```

```
      }
    >>
  }
>>
```

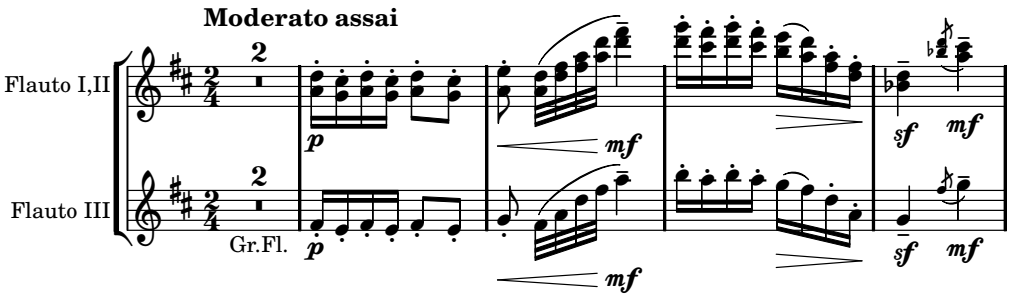


Observe, además, que debe añadir acordes (paréntesis en ángulo < >) alrededor de cada una de las instrucciones `\parenthesize`.

Véase también

Fragmentos de código: Sección "Percussion" in *Fragmentos de código*.

2.6 Instrumentos de viento



Esta sección incluye ciertos elementos de notación musical que afloran al escribir para instrumentos de viento.

2.6.1 Notación común para instrumentos de viento

Esta sección trata algunos asuntos que son comunes a casi todos los instrumentos de viento.

Referencias para instrumentos de viento

Muchas cuestiones de la notación para instrumentos de viento tienen que ver con las respiraciones y los golpes de lengua:

- Las respiraciones se pueden especificar mediante silencios o marcas de respiración, véase [Marcas de respiración], página 94.
- La ejecución ligada se indica mediante ligaduras de expresión, véase [Ligaduras de expresión], página 92.
- Los distintos golpes de lengua, desde legato hasta staccato pasando por non legato, se presentan por lo general mediante articulaciones, en ocasiones combinadas con ligaduras de expresión, véase [Articulaciones y ornamentos], página 84 y Sección B.10 [Lista de articulaciones], página 513.
- El frullato se indica generalmente mediante una indicación de trémolo y una marca textual sobre la nota. Véase [Repeticiones de trémolo], página 110.

También existen otros aspectos de la notación musical que son de aplicación a los instrumentos de viento:

- Muchos instrumentos de viento son transpositores, véase [Transposición de los instrumentos], página 17.

- El portamento es característico del trombón, pero otros instrumentos de viento pueden realizar glissandos con llaves o válvulas. Véase [Glissando], página 96.
- Los glissandos de serie armónica, que son posibles en todos los metales pero bastante específicos de las trompas, se escriben en general como notas de adorno, véase [Notas de adorno], página 78.
- Las inflexiones de tono al final de una nota se tratan en [Caídas y elevaciones], página 95.
- Los golpes de llave o de válvula se suelen indicar con el estilo `cross` (aspas) de cabezas de nota, véase [Cabezas de nota especiales], página 27.
- Los instrumentos de viento-madera pueden sobreinflar las notas graves para producir armónicos. Éstos se muestran por medio de la articulación `flageolet`. Véase Sección B.10 [Lista de articulaciones], página 513.
- El uso de sordinas para los metales se suele indicar mediante marcas de texto, pero en los lugares donde se suceden muchos cambios rápidos es mejor utilizar las articulaciones `stopped` (tapado) y `open` (abierto). Véase [Articulaciones y ornamentos], página 84 y Sección B.10 [Lista de articulaciones], página 513.
- Las trompas tapadas se indican por medio de la articulación `stopped`. Véase [Articulaciones y ornamentos], página 84.

Fragmentos de código seleccionados
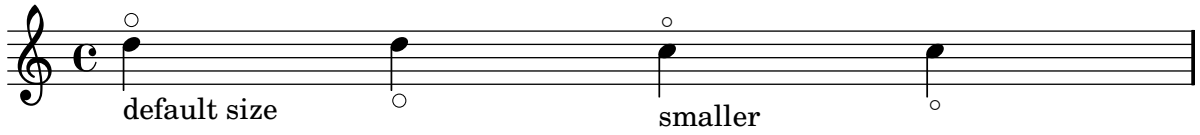
Cambiar el tamaño de la marca de \flageolet

Para hacer más pequeño el círculo de `\flageolet` (armónico) utilice la siguiente función de Scheme.

```
flageoletPequeño = #(let ((m (make-music 'ArticulationEvent
                           'articulation-type "flageolet")))
       (set! (ly:music-property m 'tweaks)
             (acons 'font-size -3
                    (ly:music-property m 'tweaks)))
       m)

\layout { ragged-right = ##f }

\relative c'' {
  d4^\flageolet_\markup { default size } d_\flageolet
  c4^\flageoletPequeño_\markup { smaller } c_\flageoletPequeño
}
```



Véase también

Referencia de la notación: [Marcas de respiración], página 94, [Ligaduras de expresión], página 92, [Articulaciones y ornamentos], página 84, Sección B.10 [Lista de articulaciones], página 513, [Repeticiones de trémolo], página 110, [Transposición de los instrumentos], página 17, [Glissando], página 96, [Notas de adorno], página 78, [Caídas y elevaciones], página 95, [Cabezas de nota especiales], página 27.

Snippets: Sección “Winds” in *Fragmentos de código*

Digitaciones

Todos los instrumentos de viento aparte del trombón requieren el uso de los distintos dedos para producir cada nota.

[REDACTAR]

2.6.2 Gaita

Esta sección contiene información adicional de utilidad para la escritura para gaita.

Definiciones para la gaita

LilyPond contiene definiciones especiales para la música de gaita escocesa de las tierras altas; para usarlas, escriba
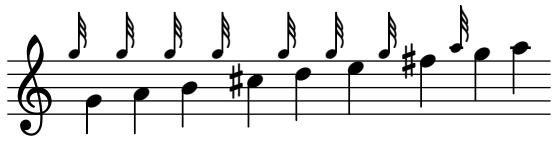
```
\include "bagpipe.ly"
```

al principio del archivo de entrada. De esta forma podrá escribir las notas de adorno especiales que son usuales en la música de gaita, mediante instrucciones breves. Por ejemplo, puede escribir `\taor` en lugar de

```
\grace { \small G32[ d G e] }
```

`bagpipe.ly` también contiene definiciones de alturas para las notas de la gaita en las octavas adecuadas, de forma que no se tenga que preocupar por `\relative` o por `\transpose`.

```
\include "bagpipe.ly"
{ \grg G4 \grg a \grg b \grg c \grg d \grg e \grg f \grA g A }
```



La música de gaita utiliza siempre la tonalidad de Re mayor (aunque esto no es completamente cierto). Sin embargo, puesto que es la única tonalidad que se puede utilizar, normalmente la armadura no se escribe. Para configurar esto de forma correcta, comience la música siempre con `\hideKeySignature` (ocultar armadura). Si, por algún motivo, quiere mostrar la armadura, puede usar `\showKeySignature` en su lugar.

Alguna música moderna utiliza digitaciones cruzadas sobre el Do y el Fa para bemolizar estas notas. Se puede indicar esto mediante `cflat` o `fflat`. De forma parecida, el sol agudo piobaireachd se puede escribir `gflat` cuando aparece en música ligera.

Véase también

Fragmentos de código: Sección “Winds” in *Fragmentos de código*.

Ejemplo de música de gaita

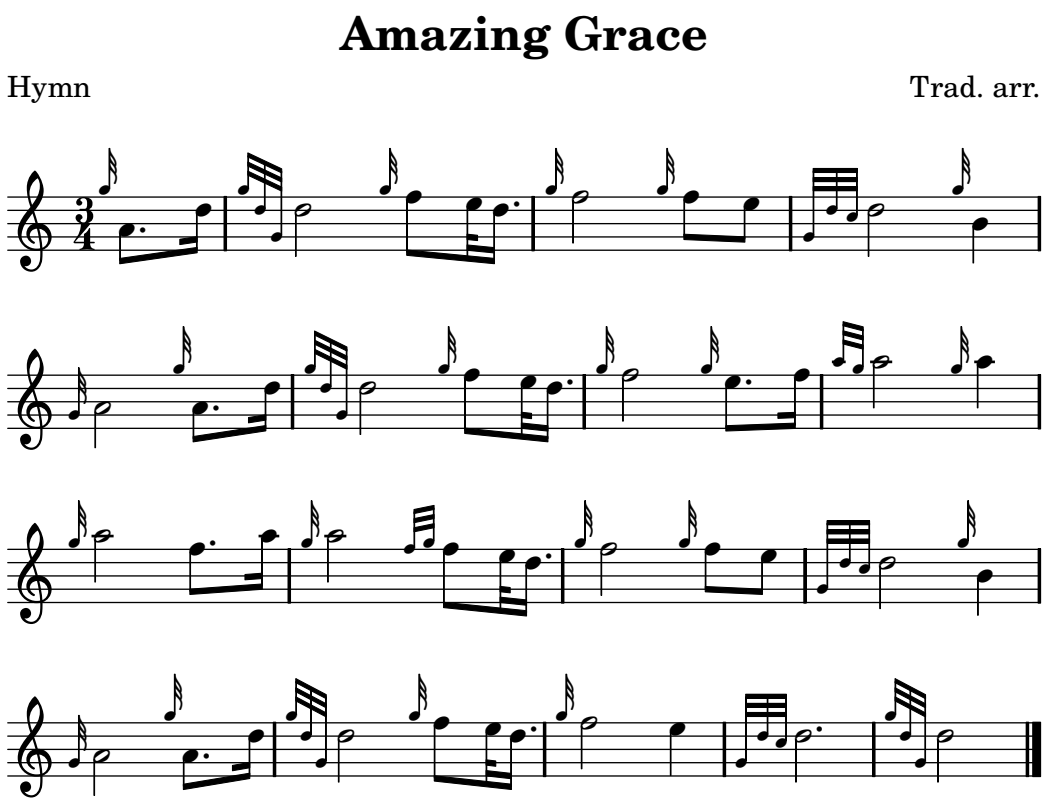
La conocida melodía Amazing Grace tiene este aspecto en notación de gaita.

```
\include "bagpipe.ly"
\layout {
  indent = 0.0\cm
  \context { \Score \remove "Bar_number_engraver" }
}

\header {
  title = "Amazing Grace"
  meter = "Hymn"
```

```
  arranger = "Trad. arr."
}

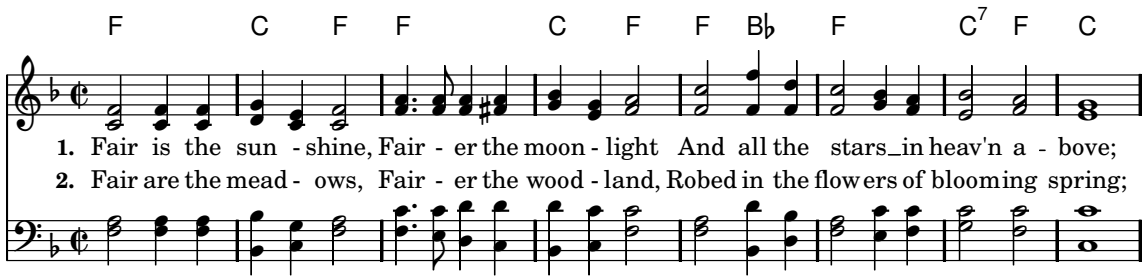
{
  \hideKeySignature
  \time 3/4
  \grg \partial 4 a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg e8. f16
  \dblA A2 \grg A4
  \grg A2 f8. A16
  \grg A2 \hdblf f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 e4
  \thrwd d2.
  \slurd d2
  \bar "|."
}
```



Véase también

Fragmentos de código: Sección “Winds” in *Fragmentos de código*.

2.7 Notación de acordes



Los acordes se pueden escribir en el modo de acordes, que reconoce ciertas convenciones europeas tradicionales de nomenclatura de acordes. También se pueden imprimir los nombres de los acordes. Además se puede imprimir notación de bajo cifrado.

2.7.1 Modo de acordes

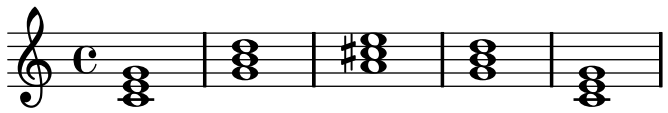
Para introducir acordes se utiliza el modo de acordes. Se usa un indicador de la estructura del acorde en lugar de la altura de sus notas.

Panorámica del modo de acordes

Los acordes se pueden escribir como música simultánea, como se explica en [Notas en acorde], página 111.

También se pueden escribir los acordes en el "modo de acordes", que es un modo de entrada que se centra en las estructuras de los acordes dentro de la música europea tradicional (donde se conoce como «cifrado americano») más que en las notas concretas. Esto es muy práctico para los que están familiarizados con la utilización de nombres para describir los acordes. Hay más información sobre los distintos modos de entrada en Sección 5.4.1 [Modos de entrada], página 406.

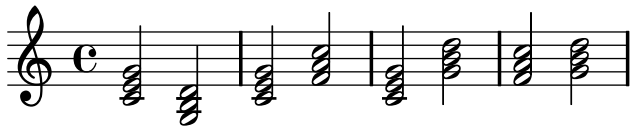
```
\chordmode { c1 g a g c }
```



Los acordes escritos con el modo de acordes son elementos musicales, y se pueden transportar igual que los acordes escritos mediante construcciones de música simultánea.

Se pueden mezclar los modos de acorde y de nota en la música secuencial:

```
<c e g>2 <g b d>
\chordmode { c2 f }
<c e g>2 <g' b d>
\chordmode { f2 g }
```



Véase también

Glossario musical: Sección "acorde" in *Glosario Musical*.
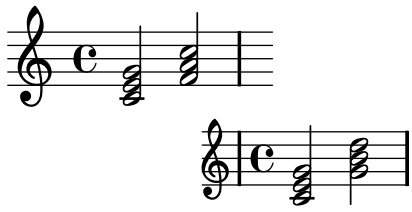
Referencia de la notación: [Notas en acorde], página 111, Sección 5.4.1 [Modos de entrada], página 406.

Fragmentos de código: Sección "Chords" in *Fragmentos de código*

Advertencias y problemas conocidos

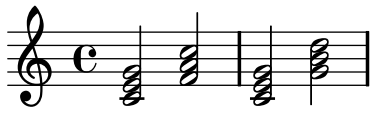
Cuando se mezclan los modos de acorde y de nota en música secuencial, y el modo de acordes aparece en primer lugar, el modo de notas crea un nuevo contexto de `Staff`.

```
\chordmode { c2 f }
<c e g>2 <g' b d>
```



Para evitar este comportamiento podemos crear explícitamente el contexto de `Staff`:

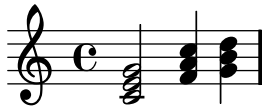
```
\new Staff {
  \chordmode { c2 f }
  <c e g>2  <g' b d>
}
```



Acordes más usuales
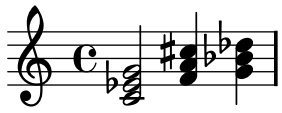
Las tríadas mayores se introducen escribiendo la fundamental y una duración opcional:

```
\chordmode { c2 f4 g }
```



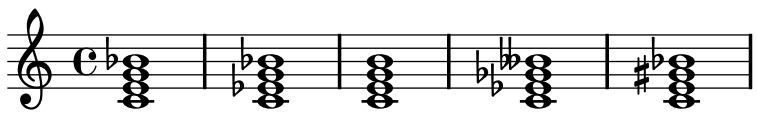
Las tríadas menores, aumentadas y disminuidas se escriben poniendo `:` y una cadena modificadora de variante después de la duración:

```
\chordmode { c2:m f4:aug g:dim }
```



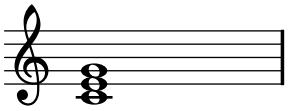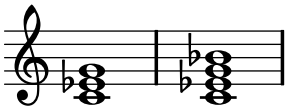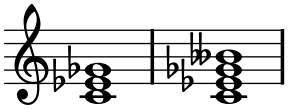
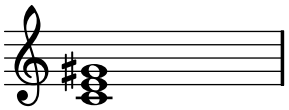
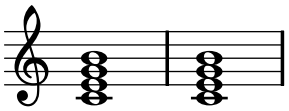
Se pueden crear acordes de séptima:

```
\chordmode { c1:7 c:m7 c:maj7 c:dim7 c:aug7 }
```



La tabla que aparece más abajo muestra el efecto de los modificadores de variante sobre los acordes de tríada y de séptima. La séptima añadida de forma predeterminada a los acordes es menor, lo que hace del acorde de séptima dominante el acorde básico de séptima. Todas las

alteraciones se consideran en relación con la séptima dominante. Hay una tabla más completa donde aparecen los usos de los modificadores en Sección B.2 [Modificadores de acorde más usuales], página 453.

Modificador	**Acción**	**Ejemplo**
Ninguno	Acción predeterminada; produce una tríada mayor.	
m, m7	Acorde menor. Este modificador baja la tercera.	
dim, dim7	Acorde disminuido. Este modificador baja la tercera, la quinta y, si existe, la séptima.	
aug	Acorde aumentado. Este modificador eleva la quinta.	
maj, maj7	Acorde de séptima mayor. Este modificador añade una séptima elevada. El `7` que sigue a `maj` es opcional. NO utilice este modificador para crear una tríada mayor.	

Véase también

Referencia de la notación: Sección B.2 [Modificadores de acorde más usuales], página 453, [Acordes extendidos y alterados], página 266.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

Advertencias y problemas conocidos

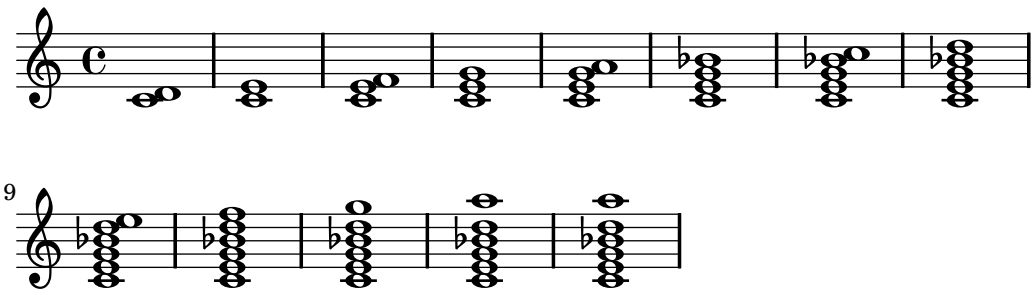
Sólo se puede usar un modificador por cada acorde, normalmente sobre la nota más aguda del mismo. Los acordes con maś de un modificador se analizan sin producir errores ni advertencias, pero el resultado es impredecible. Los acordes que no se pueden conseguir con un solo modificador se deben alterar en sus notas individuales como se describe en [Acordes extendidos y alterados], página 266.

Acordes extendidos y alterados

Se pueden crear estructuras de acorde de complejidad arbitraria dentro del modo de acordes. Se puede usar la cadena modificadora para extender un acorde, añadir o quitar notas, elevar o bajar notas del acorde y añadir un bajo distinto o crear una inversión.

El primer número que sigue al : se considera que es el ámbito del acorde. El acorde se construye secuencialmente añadiendo terceras a la fundamental hasta que se alcanza el número especificado. Observe que la séptima añadida como parte de un acorde extendido es la séptima menor, no mayor. Si el ámbito no es una tercera (p.ej. 6), se añaden terceras hasta la tercera más alta inferior al ámbito, y después se añade la nota del ámbito. El mayor valor posible para el ámbito es 13. Cualquier valor mayor se interpreta como 13.

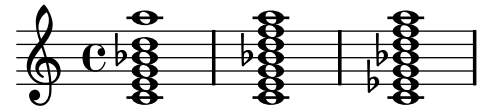
```
\chordmode {
  c1:2 c:3 c:4 c:5
  c1:6 c:7 c:8 c:9
  c1:10 c:11 c:12 c:13
  c1:14
}
```



Observe que `c:5` es idéntico a `c` (los dos producen una tríada de Do mayor).

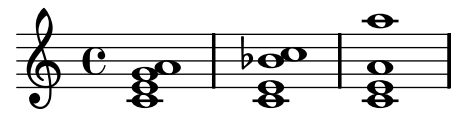
Puesto que una oncena sin alteración no suena bien cuando se combina con una trecena sin alteración, se elimina la oncena de los acordes de `:13` (a no ser que se añada explícitamente).

```
\chordmode {
  c1:13 c:13.11 c:m13
}
```



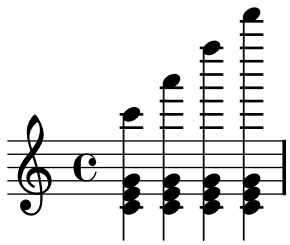
Se pueden añadir notas individuales a un acorde. Las adiciones siguen el ámbito y van prefijadas por un punto (`.`). La séptima normal que se añade a un acorde es la séptima menor, no mayor.

```
\chordmode {
  c1:5.6 c:3.7.8 c:3.6.13
}
```
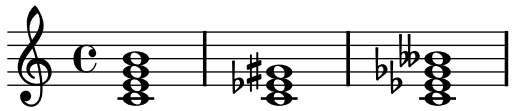


Las notas añadidas pueden ser tan agudas como se desee.

```
\chordmode {
  c4:5.15 c:5.20 c:5.25 c:5.30
}
```



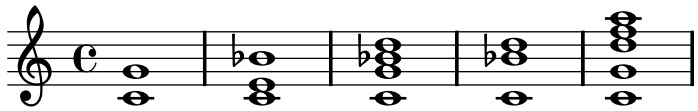
Las notas añadidas a los acordes se pueden alterar mediante la adición de los sufijos `-` o `+` al número. Para alterar una nota que se ha incluido automáticamente como parte de la estructura básica del acorde, añádala como una nota alterada.

```
\chordmode {
  c1:7+ c:5+.3- c:3-.5-.7-
}
```



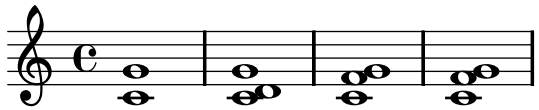
Una nota que se quiere suprimir de un acorde se indica dentro de la cadena modificadora prefijándola por un acento circunflejo `^`. Sólo se permite una supresión con `^` dentro de una cadena modificadora.

```
\chordmode {
  c1^3 c:7^5 c:9^3 c:9^3.5 c:13.11^3.7
}
```



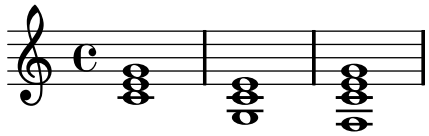
Se puede añadir el modificador `sus` a la cadena modificadora para crear acordes suspendidos. Esto elimina la tercera del acorde. Esciba bien `2` o bien `4` para añadir la seguna o la cuarta al acorde, respectivamente. `sus` equivale a `^3`; `sus4` equivale a `.4^3`.

```
\chordmode {
  c1:sus c:sus2 c:sus4 c:5.4^3
}
```



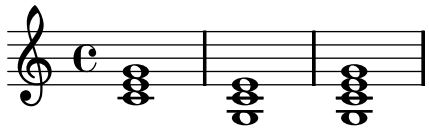
Se pueden especificar las inversiones (colocar en el bajo una nota distinta de la fundamental del acorde) y notas de bajo añadidas mediante la adición de */nota* al acorde.

```
\chordmode {
  c1 c/g c/f
}
```



Se puede añadir una nota del bajo que forma parte del acorde, en lugar de ser movida por efecto de una inversión, mediante el uso de `/+`*nota*.

```
\chordmode {
  c1 c/g c/+g
}
```



Los modificadores de acorde que se pueden utilizar para producir una amplia variedad de acordes estándar se muestran en Sección B.2 [Modificadores de acorde más usuales], página 453.

Véase también

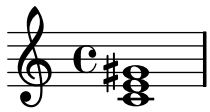
Referencia de la notación: Sección B.2 [Modificadores de acorde más usuales], página 453.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

Advertencias y problemas conocidos

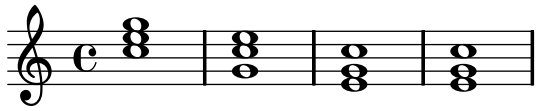
Cada nota sólo puede estar presente en el acorde una sola vez. Lo que sigue produce simplemente el acorde aumentado, porque `5+` se interpreta en último lugar.

```
\chordmode { c1:5.5-.5+ }
```



Sólo se puede crear la primera inversión mediante la adición de un bajo. La segunda inversión requiere cambiar la fundamental del acorde.

```
\chordmode {
  c'1: c':/g e:6-3-^5 e:m6-^5
}
```



2.7.2 Imprimir los acordes

Los acordes se pueden imprimir por su nombre, además de la impresión estándar como notas sobre un pentagrama.

Impresión de los nombres de acorde

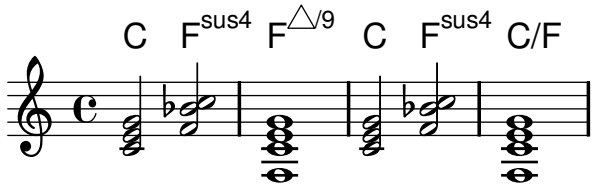
Los nombres de acorde se imprimen dentro del contexto `ChordNames`:

```
\new ChordNames {
  \chordmode {
    c2 f4. g8
  }
}
```



Los acordes se pueden escribir como notas simultáneas o a través del uso del modo de acordes. El nombre de acorde que se imprime es el mismo independientemente del modo de entrada, a no ser que existan inversiones o bajos añadidos:

```
<<
  \new ChordNames {
    <c e g>2  <f bes c>
    <f c' e g>1
    \chordmode {
      c2 f:sus4 c1:/f
    }
  }
  {
    <c e g>2  <f bes c>
    <f, c' e g>1
    \chordmode {
      c2 f:sus4 c1:/f
    }
  }
>>
```



`\chords { ... }` es una forma abreviada de escribir `\new ChordNames { \chordmode { ... } }`.

```
\chords {
  c2 f4.:m g8:maj7
}
```




```
\new ChordNames {
  \chordmode {
    c2 f4.:m g8:maj7
  }
}
```



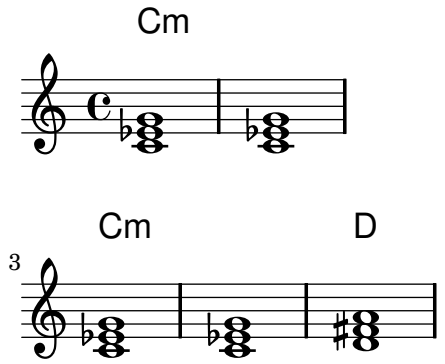
Fragmentos de código seleccionados

Imprimir los acordes cuando se produce un cambio

Se pueden imprimir los acordes exclusivamente al comienzo de las líneas y cuando cambia el acorde.

```
armonias = \chordmode {
  c1:m c:m \break c:m c:m d
}
<<
  \new ChordNames {
    \set chordChanges = ##t
```
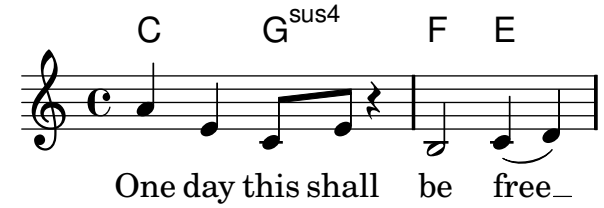
```
    \armonias
  }
  \new Staff {
    \relative c' { \armonias }
  }
>>
```



Hoja guía de acordes o «lead sheet» sencilla

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet»:

```
<<
  \chords { c2 g:sus4 f e }
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Véase también

Glosario musical: Sección “acorde” in *Glosario Musical*.

Referencia de la notación: [Escribir música en paralelo], página 123.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ChordNames” in *Referencia de Funcionamiento Interno*, Sección “ChordName” in *Referencia de Funcionamiento Interno*, Sección “Chord_name_engraver” in *Referencia de Funcionamiento Interno*, Sección “Volta_engraver” in *Referencia de Funcionamiento Interno*, Sección “Bar_engraver” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

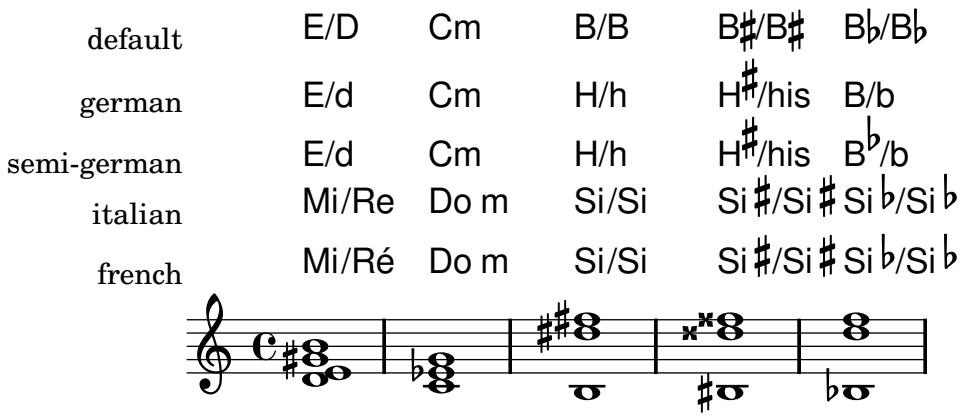
Los acordes que contienen inversiones o bajos alterados no reciben un nombre adecuado si se escriben usando música simultánea.

Nombres de acorde personalizados

No hay un sistema único para nombrar acordes. Las distintas tradiciones musicales usan distintos nombres para el mismo conjunto de acordes. También hay distintos símbolos impresos para un nombre de acorde dado. Los nombres y símbolos que se imprimen para los nombres de acorde dados se pueden personalizar.

El esquema de nomenclatura de acordes predeterminado es un sistema para música de Jazz, propuesto por Klaus Ignatzek (véase Apéndice A [Lista bibliográfica], página 451). También funcionan otros dos esquemas de nomenclatura de acordes: una notación de acordes de Jazz alternativa y un esquema sistemático llamado Acordes de Banter. La notación de Jazz alternativa también se puede ver en el cuadro Sección B.1 [Carta de nombres de acordes], página 452.

Además de los distintos sistemas de nomenclatura, se usan diferentes nombres de nota para la fundamental en los distintos idiomas. Las variables predefinidas `\germanChords`, `\semiGermanChords`, `\italianChords` y `\frenchChords` establecen el valor de estas variables. El efecto se muestra aquí:



Si ninguno de los ajustes predeterminados proporciona el resultado deseado, se puede afinar la presentación de los nombres de acorde a través de las siguientes propiedades.

`chordRootNamer`
: El nombre de acorde se imprime normalmente como una letra para la fundamental con una alteración opcional. La transformación de la nota en la letra se realiza por parte de esta función. Los nombres de nota especiales (por ejemplo, la 'H' alemana para un acorde de Si) se pueden producir almacenando una cunción nueva en esta propiedad.

`majorSevenSymbol`
: Esta propiedad contiene el objeto de marcado que se usa para hacer el seguimiento de la salida de `chordRootNamer` para identificar un acorde de séptima mayor. Las opciones predefinidas son `whiteTriangleMarkup` (triángulo blanco) y `blackTriangleMarkup` (triángulo negro).

`chordNoteNamer`
: Cuando el nombre del acorde contiene notas adicionales aparte de la fundamental (p.ej., un bajo añadido), se utiliza esta función para imprimir la nota adicional. De forma predeterminada se imprime la nota usando `chordRootNamer`. La propiedad `chordNoteNamer` se puede establecer a una función especializada para cambiar este comportamiento. Por ejemplo, el bajo se puede imprimir en minúscula.

`chordNameSeparator`
: Las diferentes partes del nombre de un acorde se separan normalmente mediante una barra inclinada. Mediante el ajuste de `chordNameSeparator`, podemos usar cualquier marcado que deseemos como separador.

`chordNameExceptions`
: Esta propiedad es una lista de parejas. El primer elemento de cada pareja es un conjunto de notas utilizadas para identificar los elementos presentes en el acorde. El segundo elemento es un elemento de marcado que sigue a la salida de `chordRootNamer` para crear el nombre del acorde.

`chordPrefixSpacer`
: La 'm' de los acordes menores se imprime por lo general imediatamente a la derecha de la fundamental. Se puede insertar un símbolo separador entre la fundamental y 'm' estableciendo `chordPrefixSpacer`. El símbolo separador no se usa cuando el acorde está alterado.

Instrucciones predefinidas

`\whiteTriangleMarkup`, `\blackTriangleMarkup`, `\germanChords`, `\semiGermanChords`, `\italianChords`, `\frenchChords`.

Fragmentos de código seleccionados

Excepciones para los nombres de acorde

Se puede usar la propiedad `chordNameExceptions` para almacenar una lista de notaciones espaciales para acordes específicos.

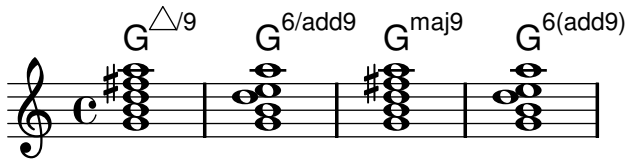
```
% cambiar maj9 y 6(add9)
% la Música de Excepciones son acordes con marcados
cambiaMusicaExcepciones = {
  <c e g b d'>1-\markup { \super "maj9" }
  <c e g a d'>1-\markup { \super "6(add9)" }
}

% Convertir la música en una lista y añadirla a las excepciones existentes.
cambiaExcepciones = #( append
  ( sequential-music-to-chord-exceptions cambiaMusicaExcepciones #t)
  ignatzekExceptions)

laMusica = \chordmode {
  g1:maj9 g1:6.9
  \set chordNameExceptions = #cambiaExcepciones
  g1:maj9 g1:6.9
}

\layout {
  ragged-right = ##t
}

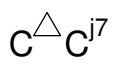
<< \context ChordNames \laMusica
   \context Voice \laMusica
>>
```



chord name major7

The layout of the major 7 can be tuned with `majorSevenSymbol`.

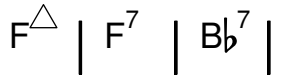
```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```



Añadir barras de compás al contexto de nombres de acorde (ChordNames)

Para añadir indicaciones de línea divisoria dentro del contexto de los nombres de acorde `ChordNames`, incluya el grabador `Bar_engraver`.

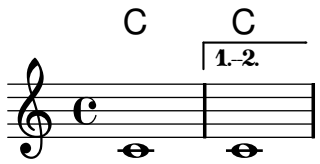
```
\new ChordNames \with {
  \override BarLine #'bar-size = #4
  \consists "Bar_engraver"
}
\chordmode {
  f1:maj7 f:7 bes:7
}
```



Corchetes de primera y segunda vez debajo de los acordes

Mediante la adición del grabador `Volta_engraver` al pentagrama pertinente, se pueden poner los corchetes de primera y segunda vez debajo de los acordes.

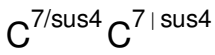
```
\score {
  <<
    \chords {
      c1
      c1
    }
    \new Staff \with {
      \consists "Volta_engraver"
    }
    {
      \repeat volta 2 { c'1 }
      \alternative { c' }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



Modificación del separador de acordes

Se puede establecer el separador entre las distintas partes del nombre de un acorde para que sea cualquier elemento de marcado.

```
\chords {
  c:7sus4
  \set chordNameSeparator
    = \markup { \typewriter | }
  c:7sus4
}
```



Véase también

Referencia de la notación: Sección B.1 [Carta de nombres de acordes], página 452, Sección B.2 [Modificadores de acorde más usuales], página 453.

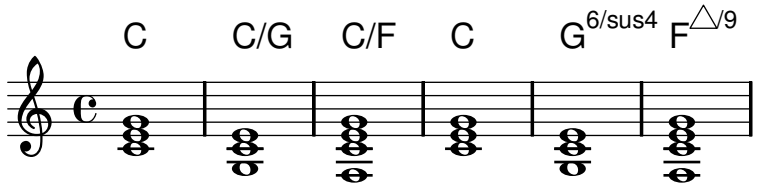
Archivos instalados: ‘`scm/chords-ignatzek.scm`’, ‘`scm/chord-entry.scm`’, ‘`ly/chord-modifier-init.ly`’.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.
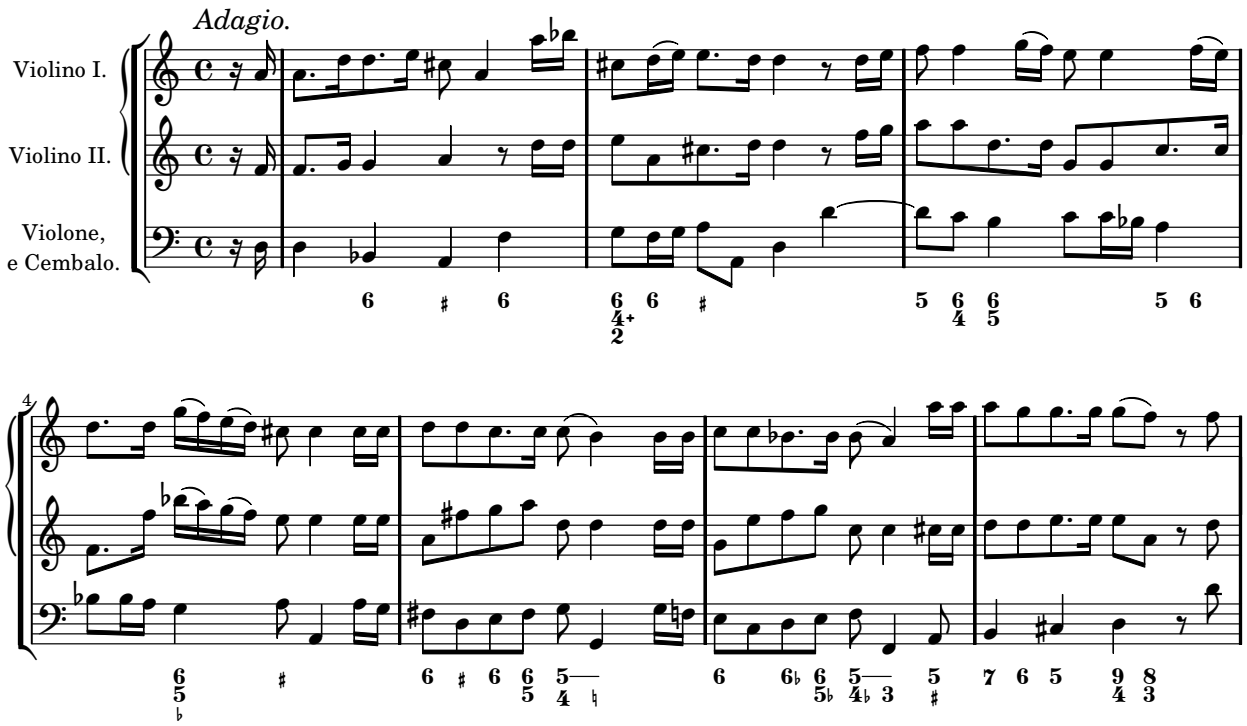
Advertencias y problemas conocidos

Los nombres de acorde se determinan a partir de las notas que están presentes en el mismo y de la información acerca de la estructura del acorde que ha podido ser introducida en `\chordmode`. Si el método de las notas simultáneas se usa para introducir los acordes, pueden resultar nombres no deseados a partir de las inversiones o de las notas del bajo.

```
misAcordes = \relative c' {
  \chordmode { c1 c/g c/f }
  <c e g>1 <g c e> <f c' e g>
}
<<
  \new ChordNames { \misAcordes }
  \new Staff { \misAcordes }
>>
```



2.7.3 Bajo cifrado

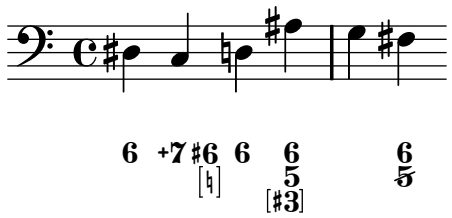


Se puede imprimir notación de bajo cifrado.

Introducción al bajo cifrado

LilyPond contempla el bajo cifrado, también conocido como basso continuo:

```
<<
  \new Voice { \clef bass dis4 c d ais g fis}
  \new FiguredBass {
    \figuremode {
      < 6 >4 < 7\+ >8 < 6+ [_!] >
      < 6 >4 <6 5 [3+] >
      < _ >4 < 6 5/>4
    }
  }
>>
```



El soporte para bajo cifrado se compone de dos partes: hay un modo de entrada, presentado por `\figuremode`, que admite la entrada de las cifras del bajo, y hay un contexto llamado `FiguredBass` que se ocupa de la impresión de los objetos del tipo `BassFigure`. El bajo cifrado también se puede imprimir dentro de contextos de `Staff`.

`\figures{ ... }` es una notación abreviada de `\new FiguredBass { \figuremode { ... } }`.

Aunque el soporte para bajo cifrado puede parecerse superficialmente al de los acordes, realmente es mucho más simple. El modo `\figuremode` simplemente almacena las cifras y el contexto `FiguredBass` las imprime tal y como se han introducido. No hay ninguna conversión a notas.

Véase también

Glosario musical: Sección “bajo cifrado” in *Glosario Musical*.

Fragmentos de código: Sección “Chords” in *Fragmentos de código*

Introducir el bajo cifrado

Se usa `\figuremode` para cambiar el mod de entrada al modo de cifras. Hay más información sobre los distintos modos de entrada en Sección 5.4.1 [Modos de entrada], página 406.

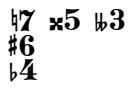
En el modo de cifras, un grupo de cifras del bajo está delimitado mediante ángulos simples, `<` y `>`. La duración se escribe después del `>`.

```
\new FiguredBass {
  \figuremode {
    <6 4>2
  }
}
```



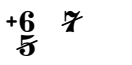
Se pueden añadir alteraciones accidentales (incluso becuadros) a las cifras:

```
\figures {
  <7! 6+ 4-> <5++> <3-->
}
```



Se pueden indicar elementos del acorde aumentados y disminuidos:

```
\figures {
  <6\+ 5/> <7/>
}
```



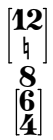
Se puede crear una barra invertida cruzando una cifra (usada normalmente para sextas elevadas):

```
\figures {
  <6> <6\\>
}
```



Se pueden incluir espacios verticales y corchetes en las cifras:

```
\figures {
  <[12 _!] 8 [6  4]>
}
```



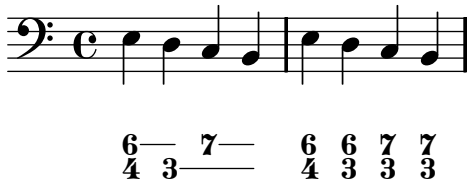
Se puede insertar cualquier elemento de marcado de texto como una cifra:

```
\figures {
  <\markup { \tiny \number 6 \super (1) } 5>
}
```



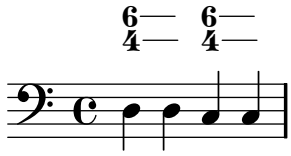
Se pueden usar líneas de continuación para indicar cifras que se repiten:

```
<<
  {
    \clef bass
    e4 d c b,
    e4 d c b,
  }
  \figures {
    \bassFigureExtendersOn
    <6 4>4 <6 3> <7 3> <7 3>
    \bassFigureExtendersOff
    <6 4>4 <6 3> <7 3> <7 3>
  }
>>
```

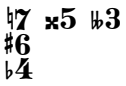
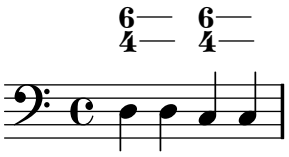


En este caso, las líneas de extensión sustituyen a las cifras existentes, a no ser que las líneas de continuación hayan sido terminadas explícitamente.

```
<<
  \figures {
    \bassFigureExtendersOn
    <6 4>4 <6 4> <6\! 4\!> <6 4>
  }
  {
    \clef bass
    d4 d c c
  }
>>
```



La siguiente tabla resume los modificadores de cifrado disponibles.

Modificador	Propósito	Ejemplo
+, -, !	Alteraciones	
\+, /	Aumentaciones y disminuciones	
\\	Sexta elevada	
\!	Fin de línea de continuación	

Instrucciones predefinidas

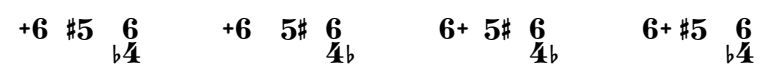
`\bassFigureExtendersOn`, `\bassFigureExtendersOff`.

Fragmentos de código seleccionados

Cambiar las posiciones de las alteraciones del bajo cifrado

Las alteraciones y los signos «más» pueden aparecer antes o después de los números, según el valor de las propiedades `figuredBassAlterationDirection` y `figuredBassPlusDirection`.

```
\figures {
  <6\+> <5+> <6 4-> r
  \set figuredBassAlterationDirection = #RIGHT
  <6\+> <5+> <6 4-> r
  \set figuredBassPlusDirection = #RIGHT
  <6\+> <5+> <6 4-> r
  \set figuredBassAlterationDirection = #LEFT
  <6\+> <5+> <6 4-> r
}
```



Véase también

Fragmentos de código: Sección “Chords” in *Fragmentos de código*.

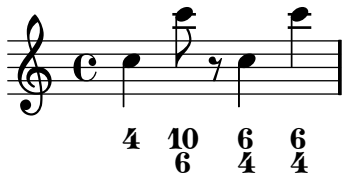
Referencia de funcionamiento interno: Sección “BassFigure” in *Referencia de Funcionamiento Interno*, Sección “BassFigureAlignment” in *Referencia de Funcionamiento Interno*, Sección “BassFigureLine” in *Referencia de Funcionamiento Interno*, Sección “BassFigureBracket” in *Referencia de Funcionamiento Interno*, Sección “BassFigureContinuation” in *Referencia de Funcionamiento Interno*, Sección “FiguredBass” in *Referencia de Funcionamiento Interno*.

Imprimir el bajo cifrado

El bajo cifrado se puede imprimir usando el contexto `FiguredBass`, o dentro de la mayoría de los contextos de pentagrama.

Cuando se presenta dentro de un contexto de `FiguredBass`, la ubicación vertical de las cifras es independiente de las notas que están en el pentagrama.

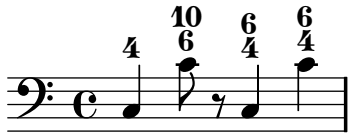
```
<<
  \relative c'' {
    c4 c'8 r8 c,4 c'
  }
  \new FiguredBass {
    \figuremode {
      <4>4 <10 6>8 s8
      <6 4>4 <6 4>
    }
  }
>>
```



En el ejemplo de arriba, el contexto `FiguredBass` se debe instanciar explícitamente para evitar crear un segundo pentagrama vacío.

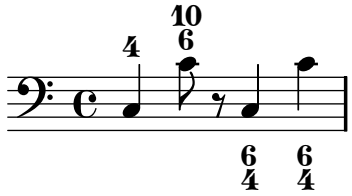
El bajo cifrado se puede añadir también a contextos de `Staff` directamente. En este caso, la posición vertical de las cifras se ajusta automáticamente.

```
<<
  \new Staff = myStaff
  \figuremode {
    <4>4 <10 6>8 s8
    <6 4>4 <6 4>
  }
  %% Poner las notas en el mismo pentagrama que los números
  \context Staff = myStaff
  {
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>
```



Cuando se escribe dentro de un contexto de pentagrama, el bajo cifrado se puede presentar encima o debajo del mismo.

```
<<
  \new Staff = myStaff
  \figuremode {
    <4>4 <10 6>8 s8
    \bassFigureStaffAlignmentDown
    <6 4>4 <6 4>
  }
  %% Poner las notas en el mismo pentagrama que los números
  \context Staff = myStaff
  {
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>
```



Instrucciones predefinidas

`\bassFigureStaffAlignmentDown`, `\bassFigureStaffAlignmentUp`, `\bassFigureStaffAlignmentNeutral`.

Véase también

Snippets: Sección “Chords” in *Fragmentos de código*.
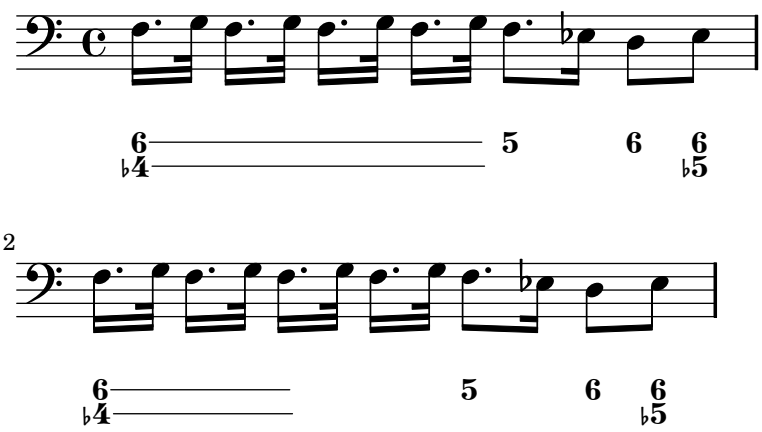
Referencia de funcionamiento interno: Sección “BassFigure” in *Referencia de Funcionamiento Interno*, Sección “BassFigureAlignment” in *Referencia de Funcionamiento Interno*, Sección “BassFigureLine” in *Referencia de Funcionamiento Interno*, Sección “BassFigureBracket” in *Referencia de Funcionamiento Interno*, Sección “BassFigureContinuation” in *Referencia de Funcionamiento Interno*, Sección “FiguredBass” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Para asegurar que las líneas extensoras funcionan adecuadamente, lo más seguro es utilizar las mismas duraciones en la línea de cifras y en la línea del bajo.

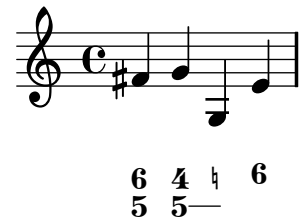
```
<<
  {
    \clef bass
    \repeat unfold 4 { f16. g32 } f8. es16 d8 es
  }
  \figures {
    \bassFigureExtendersOn
    % Los extensores aquí son correctos, con el mismo ritmo que el bajo
    \repeat unfold 4 { <6 4->16. <6 4->32 }
```

```
    <5>8. r16 <6>8 <6\! 5->
  }
>>
<<
  {
    \clef bass
    \repeat unfold 4 { f16. g32 } f8. es16 d8 es
  }
  \figures {
    \bassFigureExtendersOn
    % Los extensores aquí son incorrectos, aunque la medida es la misma
    <6 4->4 <6 4->4
    <5>8. r16 <6>8 <6\! 5->
  }
>>
```



Si se usan líneas extensoras, las cifras adyacentes con el mismo número en distinta colocación puede hacer que se inviertan las posiciones de las cifras.

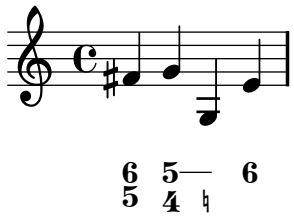
```
<<
 { fis4 g g, e' }
  \figures {
    \bassFigureExtendersOn
    <6 5>4 <5\! 4> < 5 _!> <6>
  }
>>
```
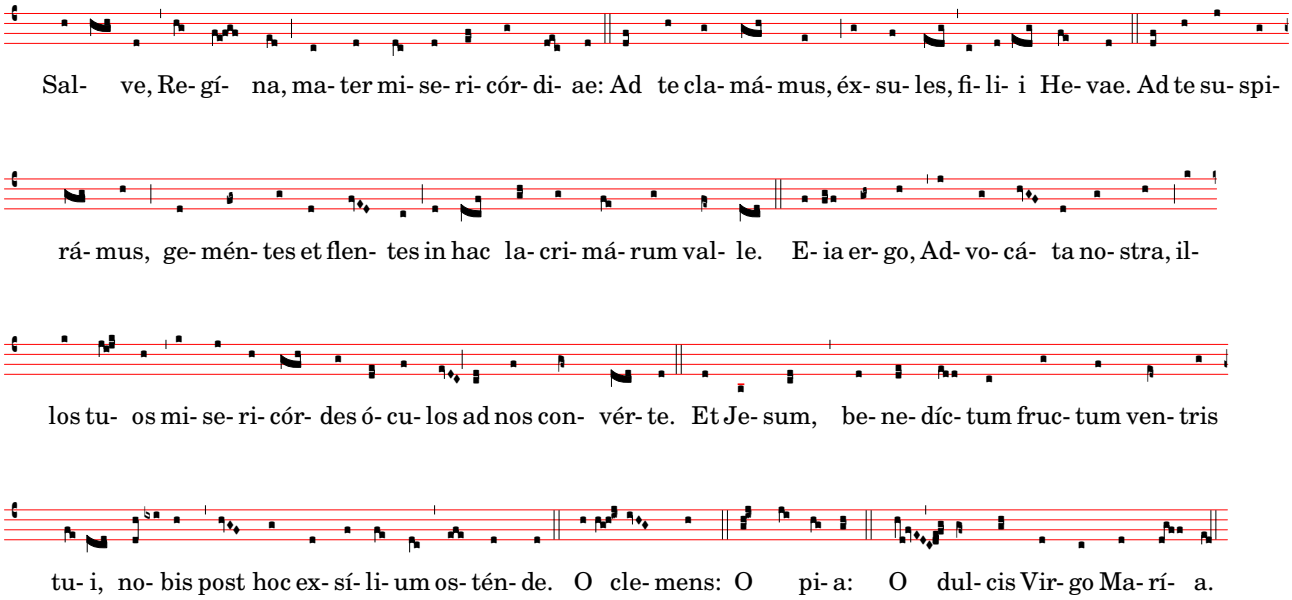


Para evitar este problema, sencillamente habilite los extensores después de la cifra y quítelos al final de la línea extensora.

```
<<
 { fis4 g g, e' }
  \figures {
```

```
    <6 5>4 <5 4>
    \bassFigureExtendersOn
    < 5 _!>4 <6>
    \bassFigureExtendersOff
  }
>>
```



2.8 Notación antigua



Entre las formas en que está contemplada la notación antigua se encuentra la posibilidad de tipografiar notación mensural y canto gregoriano. Se puede acceder a estas funcionalidades mediante la modificación de propiedades de estilo de objetos gráficos como la cabeza de las notas o los silencios, o bien mediante la utilización de uno de los contextos predefinidos para la notación mensural o gregoriana.

Muchos objetos gráficos, como la cabeza y el corchete de las figuras, las alteraciones, la indicación de compás y los silencios ofrecen una propiedad `style` (estilo), que se puede cambiar para emular varios estilos diferentes de notación antigua. Véase

- [Cabezas de nota de la música mensural], página 289,
- [Alteraciones y armaduras de la música mensural], página 291,
- [Silencios de la música mensural], página 291,
- [Claves de la música mensural], página 287,
- [Claves de canto gregoriano], página 295,
- [Corchetes de la música mensural], página 290,
- [Indicaciones de compás de la música mensural], página 288.

Ciertos conceptos de notación se incluyen específicamente para la notación antigua:

- [Custos], página 285,
- [Divisiones], página 296,
- [Ligaduras], página 285.

2.8.1 Panorámica de los estilos contemplados

Están disponibles tres estilos para la composición tipográfica del canto gregoriano:

- *Editio Vaticana* es un estilo completo para canto gregoriano, que sigue la apariencia de las ediciones de Solesmes, los libros de canto oficiales del Vaticano desde 1904. Lilypond contempla todos los signos de notación que se utilizan en este estilo, entre ellos las ligaduras, *custodias* y símbolos especiales como el quilisma y el oriscus.
- El estilo *Editio Medicaea* ofrece ciertas funcionalidades que se usan en las ediciones Medicaea (o de Ratisbona) que se usaban con anterioridad a las ediciones de Solesmes. Las diferencias más significativas con respecto al estilo *Vaticana* son las claves, que tienen trazos descendentes, y la cabeza de las figuras, que son cuadradas y regulares.
- El estilo *Hufnagel* (en "uña de herradura") o *gótico* reproduce el estilo de escritura en los manuscritos de canto medievales de Alemania y Centro Europa. Recibe el nombre de la forma de la nota básica (la *virga*), que parece una pequeña uña.

Tres estilos emulan la apariencia de los manuscritos y ediciones impresas de música mensural tardomedievales y renacentistas:

- El estilo *Mensural* se asemeja bastante al estilo de escritura utilizado en los manuscritos tardomedievales y renacentistas tempranos, con la cabeza de sus figuras en forma de rombo pequeñas y estrechas, y los silencios que imitan un estilo dibujado a mano.
- El estilo *Neomensural* es una versión modernizada y estilizada del anterior: la cabeza de las figuras es más ancha y los silencios están formados por trazos rectilíneos. Este estilo está particularmente indicado, p.ej., para los *incipits* de transcripciones de fragmentos de música mensural.
- El estilo *Petrucci* recibe su nombre de Ottaviano Petrucci (1466-1539), el primer impresor en utilizar tipos móviles para la música (en su *Harmonice musices odhecaton*, 1501). Este estilo utiliza para las figuras una cabeza más grande que los otros estilos mensurales.

Baroque y *Classical* no son estilos completos pero difieren del estilo predeterminado solamente en ciertos detalles: algunas cabezas de nota (Baroque) y el silencio de negra (Classical).

Sólo el estilo mensural tiene alternativas para todos los aspectos de la notación. Así, no existen silencios ni corchetes en los estilos gregorianos, pues estos símbolos no se usan en la notación del canto llano, y el estilo Petrucci no posee corchetes ni alteraciones por sí mismo.

Cada elemento de la notación se puede cambiar independientemente de los otros, y así podemos usar corchetes mensurales, cabezas de petrucci, silencios de classical y claves de vaticana en la misma pieza, si queremos.

2.8.2 Notación antigua: funcionalidades comunes

Contextos predefinidos

Para el canto gregoriano y la notación mensural, están a nuestra disposición contextos predefinidos de voz y de pauta, que establecen todos los símbolos de notación a unos valores adecuados para estos estilos. Si nos satisfacen estos valores predeterminados, podemos a continuación introducir directamente las notas sin la necesidad de preocuparnos de los detalles sobre cómo personalizar un contexto. Véanse uno de los contextos predefinidos `VaticanaVoice`, `VaticanaStaff`, `MensuralVoice`, y `MensuralStaff`.
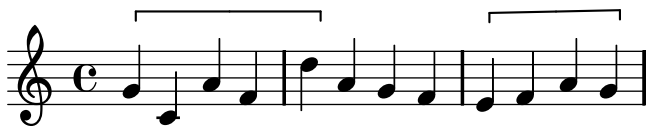
- [Contextos del canto gregoriano], página 294,
- [Contextos de la música mensural], página 286.

Ligaduras

Una ligadura es un símbolo gráfico que representa al menos dos notas distintas. Las ligaduras aparecieron originalmente en los manuscritos de la notación del canto gregoriano para denotar secuencias de notas ascendentes o descendentes en la misma sílaba. Se usan también en notación mensural.

Las ligaduras se escriben *encerrándolas* dentro de un par `\[` y `\]`. Ciertos estilos de ligadura podrían necesitar una sintaxis de entrada adicional específica para este tipo de ligadura en particular. De forma predeterminada, el grabador Sección "LigatureBracket" in *Referencia de Funcionamiento Interno* se limita a colocar un corchete recto sobre la ligadura:

```
\transpose c c' {
  \[ g c a f d' \]
  a g f
  \[ e f a g \]
}
```



Están disponibles otros dos estilos de ligaduras: el Vaticana para canto gregoriano, y el Mensural para música mensural (sólo están contempladas las ligaduras mensurales blancas para la música mensural, y con ciertas limitaciones). Para usar cualquiera de estos estilos, se debe sustituir el grabador predeterminado `Ligature_bracket_engraver` por uno de los grabadores especializados en ligaduras del contexto Sección "Voice" in *Referencia de Funcionamiento Interno*, como se explica en [Ligaduras mensurales blancas], página 292 y [Ligaduras de neumas cuadrados gregorianos], página 298.

Véase también

Advertencias y problemas conocidos

Las ligaduras necesitan un espaciado especial que aún no se ha desarrollado. Como consecuencia, la mayor parte del tiempo existe una separación excesiva entre las ligaduras, y a menudo los cortes de línea no son satisfactorios. Además, la letra de los cantos no se alinea correctamente con las ligaduras.

Las alteraciones accidentales no se deben imprimir dentro de las ligaduras, sino que deben reunirse e imprimirse juntas delante de ellas.

La sintaxis todavía utiliza el obsoleto estilo infijo `\[ expresión_musical \]`. Por motivos de consistencia, esto cambiará algún día al estilo postfijo `nota\[ ... nota\]`.

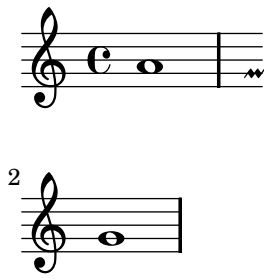
Custos

Un *custos* (en plural: *custodias*, que en latín significa "guarda") es un símbolo que aparece al final de una pauta. Anticipa la altura de la primera o primeras notas de la línea siguiente, ayudando al intérprete con los saltos de línea durante la ejecución.

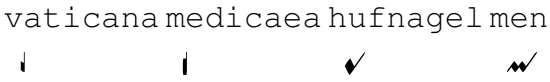
Las custodias se utilizaron mucho en la notación musical hasta el siglo XVII. En nuestros días, sobreviven sólo en algunas formas particulares de notación musical como las ediciones

contemporáneas de canto gregoriano como la *Editio Vaticana*. Hay distintos glifos para el custos que se utilizan en las diferentes variaciones de estilo notacional.

Para tipografiar las custodias, simplemente coloque un grabador Sección “Custos_engraver” in *Referencia de Funcionamiento Interno* dentro del contexto Sección “Staff” in *Referencia de Funcionamiento Interno* (el contexto del pentagrama) al declarar el bloque de `\layout` (el bloque de disposición de la página), y modifique el estilo del custos con una instrucción `\override` si lo desea, como se muestra en el ejemplo siguiente:



El glifo del custos se selecciona mediante la propiedad `style`. Los estilos contemplados son `vaticana`, `medicaea`, `hufnagel` y `mensural`. Se muestran en el siguiente fragmento:



Véase también

Referencia de funcionamiento interno: Sección “Custos” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Ancient notation” in *Fragmentos de código*.

Soporte para bajo cifrado

Está contemplada de forma limitada la notación de bajo cifrado barroco: véase Sección 2.7.3 [Bajo cifrado], página 276.

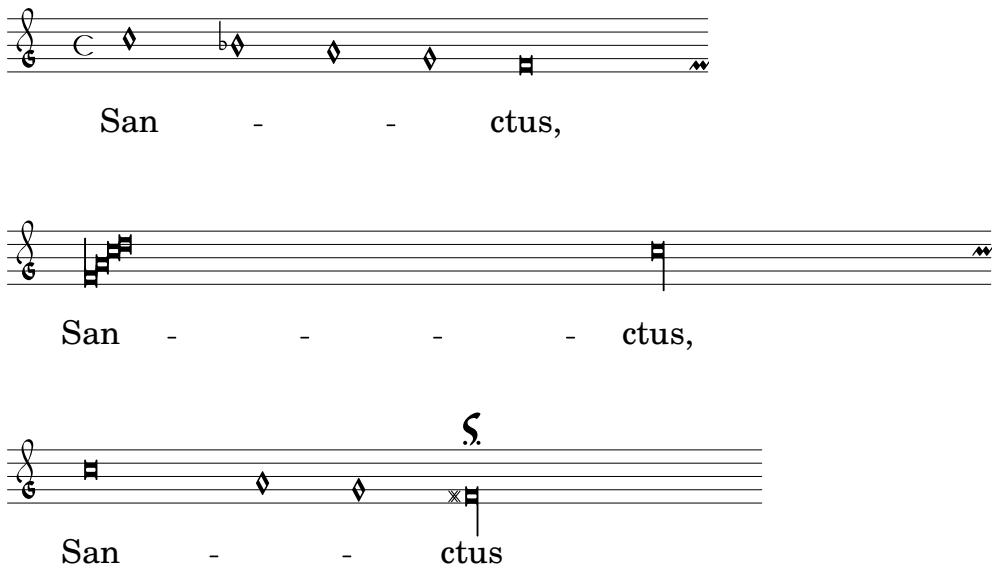
2.8.3 Tipografiar música mensural

Contextos de la música mensural

Los contextos predefinidos de voz `MensuralVoice` y de pauta `MensuralStaff` se pueden usar para grabar una pieza en estilo mensural. Estos contextos establecen a unos valores iniciales adecuados todas las propiedades relevantes de los contextos y de los objetos gráficos, de manera que pueda comenzar inmediatamente a escribir el canto, como lo demuestra el siguiente fragmento:

```
\score {
  <<
    \new MensuralVoice = "discantus" \transpose c c' {
      \override Score.BarNumber #'transparent = ##t {
        c'1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c'\breve d'\melismaEnd \]
        c'\longa
        c'\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
```

```
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```
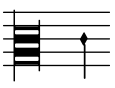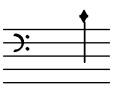


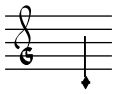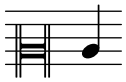
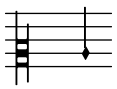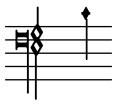
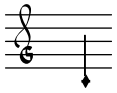
Véase también

Claves de la música mensural

La siguiente tabla presenta todas las claves antiguas que están contempladas a través de la instrucción `\clef`. Algunas de las claves usan el mismo glifo, pero se diferencian sólo en la línea en que se imprimen. En tales casos, se usa un número añadido al nombre para enumerar dichas claves, numerado desde la línea inferior hasta la superior. Aún así, puede forzar manualmente que se escriba un glifo de clave sobre una línea arbitraria, como se encuentra descrito en [Clave], página 12. La nota impresa a la derecha de cada clave en la columna de los ejemplos denota la situación del Do central (`c'`) con respecto a dicha clave.

Petrucci usaba claves de Do que tenían barras verticales en el lado izquierdo en un equilibrio variable, dependiendo de la línea de la pauta en que se imprimían.

Descripción	**Claves posibles**	**Ejemplo**
clave de Do, mensural	`mensural-c1`, `mensural-c2`, `mensural-c3`, `mensural-c4`	
clave de Fa, mensural	`mensural-f`	

clave de Sol, mensural	`mensural-g`	
clave de Do, neomensural	`neomensural-c1`, `neomensural-c2`, `neomensural-c3`, `neomensural-c4`	
claves de Do en el estilo de Petrucci, para su uso en distintas líneas de la pauta (el ejemplo muestra la clave de Do en segunda)	`petrucci-c1`, `petrucci-c2`, `petrucci-c3`, `petrucci-c4`, `petrucci-c5`	
clave de Fa en el estilo de Petrucci	`petrucci-f`	
clave de Sol en el estilo de Petrucci	`petrucci-g`	

Véase también

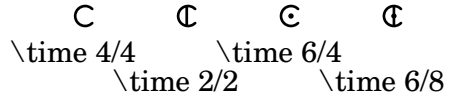
Referencia de la notación: [Clave], página 12.

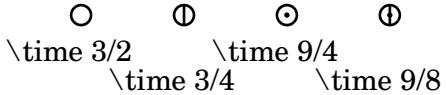
Advertencias y problemas conocidos

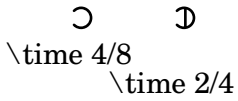
La clave de Sol mensural está asignada a la clave de Sol de Petrucci.

Indicaciones de compás de la música mensural
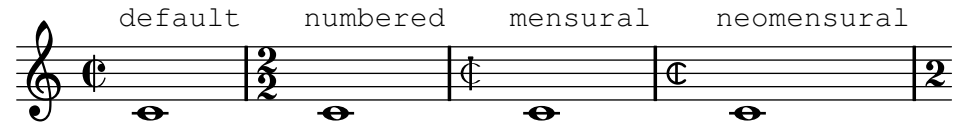
Los símbolos de mensuración (que son algo similar, pero no exactamente igual que las indicaciones de compás) están contemplados de forma limitada. Los glifos están configurados de forma fija para determinadas fracciones de tiempo. En otras palabras, para obtener un símbolo de mensuración determinado con la instrucción `\time n/m`, se tienen que escoger `n` y `m` según la siguiente tabla

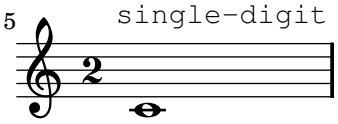






Utilice la propiedad `style` del elemento gráfico Sección "TimeSignature" in *Referencia de Funcionamiento Interno* para seleccionar las indicaciones de compás de la música antigua. Los estilos contemplados son `neomensural` y `mensural`. La tabla anterior utiliza el estilo `neomensural`. Los ejemplos siguientes muestran las diferencias que existen entre los estilos:





Véase también

Referencia de la notación: [Indicación de compás], página 46 ofrece una introducción general al uso de las indicaciones de compás.

Advertencias y problemas conocidos

Las relaciones de duración de las notas no se modifican cuando cambia la indicación de compás. Por ejemplo, la razón de 1 breve = 3 semibreves (*tempus perfectum*) se debe hacer a mano mediante el establecimiento de

```
breveTP = #(ly:make-duration -1 0 3 2)
...
{ c\breveTP f1 }
```

Esto fija un valor para `breveTP` de 3/2 multiplicado por 2 = 3 veces una redonda.

Los símbolos `mensural68alt` y `neomensural68alt` (alternativos al símbolo de 6/8) no se puede seleccionar directamente con `\time`. En su lugar, utilice `\markup {\musicglyph #"timesig.mensural68alt" }`.

Cabezas de nota de la música mensural

Para la notación antigua, se puede elegir un estilo de cabezas de notas distinto al `default` (estilo predeterminado). Esto se consigue estableciendo la propiedad `style` del objeto Sección "NoteHead" in *Referencia de Funcionamiento Interno* a los valores `baroque`, `neomensural`, `mensural` o `petrucci`.

El estilo `baroque` se diferencia del estilo predeterminado `default` en que:

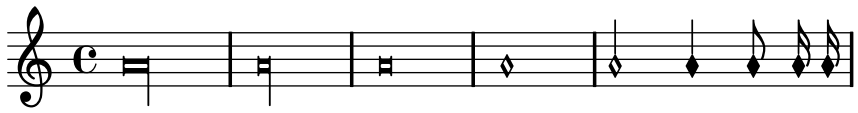
- Proporciona una cabeza de nota para la `maxima`, y
- Usa una forma cuadrada para las cabezas de `\breve`.

Los estilos `neomensural`, `mensural` y `petrucci` se diferencian del estilo `baroque` en que:

- Usan cabezas romboidales para las semibreves y todas las notas más cortas, y
- Centran las plicas sobre las cabezas.

El ejemplo siguiente muestra el estilo `petrucci`:

```
\set Score.skipBars = ##t
\autoBeamOff
\override NoteHead #'style = #'petrucci
a'\maxima a'\longa a'\breve a'1 a'2 a'4 a'8 a'16 a'
```



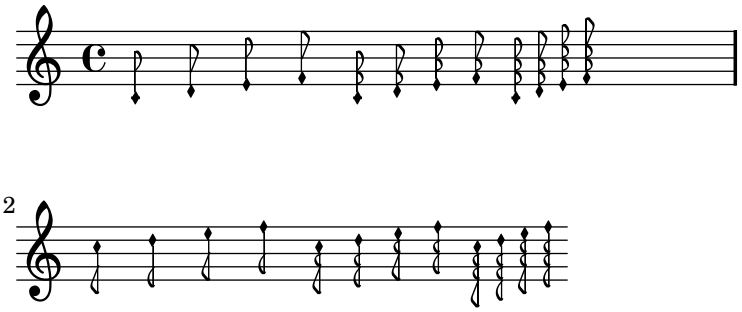
Véase también

Sección B.7 [Estilos de cabezas de nota], página 475 ofrece una visión de conjunto sobre la totalidad de los estilos de cabeza disponibles.

Corchetes de la música mensural

Utilice la propiedad `flag-style` del elemento gráfico Sección “Stem” in *Referencia de Funcionamiento Interno* para seleccionar los corchetes de estilo antiguo. Aparte del estilo de corchete por defecto `default`, sólo está contemplado el estilo `mensural`

```
\override Stem #'flag-style = #'mensural
\override Stem #'thickness = #1.0
\override NoteHead #'style = #'mensural
\autoBeamOff
c'8 d'8 e'8 f'8 c'16 d'16 e'16 f'16 c'32 d'32 e'32 f'32 s8
c''8 d''8 e''8 f''8 c''16 d''16 e''16 f''16 c''32 d''32 e''32 f''32
```



Observe que la bandera más interna de cada uno de los corchetes mensurales siempre se alinea verticalmente con una línea de la pauta.

No existe un estilo particular de corchete para la notación neo-mensural.

No existen corchetes en la notación del canto gregoriano.

Véase también

Advertencias y problemas conocidos

La unión de los corchetes antiguos a las plicas está ligeramente desviada.

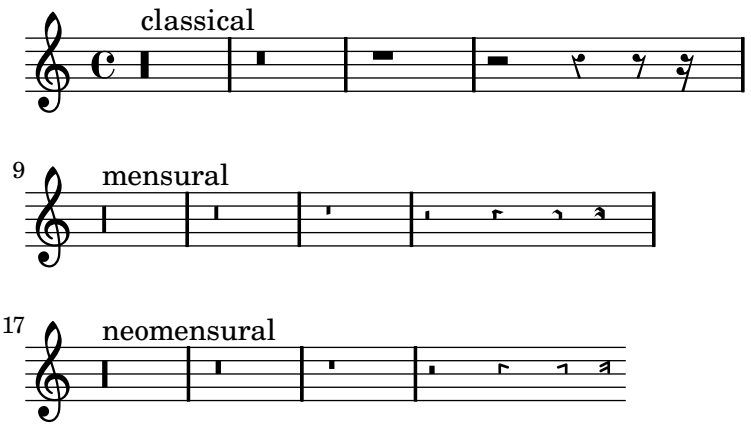
La alineación vertical de cada uno de los corchetes con una línea de la pauta da por supuesto que las plicas siempre terminan exactamente encima o bien exactamente en el medio de dos líneas de la pauta. Esto puede no ser siempre cierto cuando se utilizan las posibilidades de disposición avanzadas de la notación clásica (que, de todas formas, por lo común se encuentra fuera del ámbito de la notación mensural).

Silencios de la música mensural

Utilice la propiedad `style` del elemento gráfico Sección “Rest” in *Referencia de Funcionamiento Interno* para seleccionar silencios antiguos. Los estilos contemplados son `classical`, `neomensural` y `mensural`. `classical` se diferencia del estilo por defecto `default` solamente en que el silencio de negra parece un silencio de corchea invertido horizontalmente. Los estilos `mensural` y `neomensural` reproducen el aspecto de los silencios de los manuscritos y ediciones impresas de hasta el s.XVI.

El ejemplo siguiente muestra los estilos `mensural` y `neomensural`:

```
\set Score.skipBars = ##t
\override Rest #'style = #'classical
r\longa^"classical" r\breve r1 r2 r4 r8 r16 s \break
\override Rest #'style = #'mensural
r\longa^"mensural" r\breve r1 r2 r4 r8 r16 s \break
\override Rest #'style = #'neomensural
r\longa^"neomensural" r\breve r1 r2 r4 r8 r16
```



No existen silencios de fusa y semifusa específicos para los estilos mensural ni neo-mensural. En su lugar se tomarán los silencios del estilo predeterminado.

Consulte Sección “Ancient notation” in *Fragmentos de código* para ver un cuadro de todos los silencios.

Véase también

Referencia de la notación: [Silencios], página 39 ofrece una introducción general al uso de los silencios.

Alteraciones y armaduras de la música mensural

El estilo `mensural` ofrece unos símbolos de sostenido y bemol distintos de los del estilo predeterminado. Si se solicita, el símbolo de becuadro se extrae del estilo `vaticana`.

mensural



El estilo de las alteraciones y de la armadura de la tonalidad se controlan a través de la propiedad `glyph-name-alist` de los elementos gráficos Sección “Accidental” in *Referencia de Funcionamiento Interno* y Sección “KeySignature” in *Referencia de Funcionamiento Interno*, respectivamente; p.ej.:

```
\override Staff.Accidental #'glyph-name-alist = #alteration-mensural-glyph-name-alist
```

Véase también

Referencia de la notación: Sección 1.1 [Alturas], página 1, [Alteraciones accidentales], página 4 y [Alteraciones accidentales automáticas], página 19 proporcionan una introducción general al uso de las alteraciones. [Armadura de la tonalidad], página 15 ofrece una introducción general al uso de las armaduras de tonalidad.
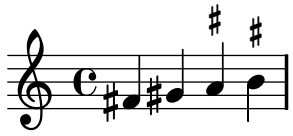
Referencia de funcionamiento interno: Sección "KeySignature" in *Referencia de Funcionamiento Interno*.

Alteraciones de anotación (*musica ficta*)

En la música europea de hasta aprox. 1600, se esperaba de los cantantes que alterasen cromáticamente ciertas notas según su propia iniciativa y de acuerdo a ciertas reglas. Esto recibe el nombre de *musica ficta*. En las transcripciones modernas, estas alteraciones accidentales se imprimen normalmente encima de la nota.
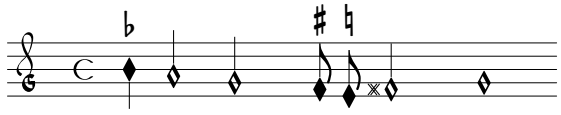
Están contempladas estas alteraciones sugeridas, y se pueden activar estableciendo `suggestAccidentals` al valor verdadero.

```
fis gis
\set suggestAccidentals = ##t
ais bis
```



Esto hará que se traten *todas* las alteraciones siguientes como *musica ficta* hasta que se desactive con `\set suggestAccidentals = ##f`. Una forma más práctica es usar `\once \set suggestAccidentals = ##t`, que puede incluso definirse como una forma breve muy conveniente:

```
ficta = { \once \set suggestAccidentals = ##t }
\score { \relative c''
  \new MensuralVoice  {
        \once \set suggestAccidentals = ##t
  bes4 a2 g2 \ficta fis8 \ficta e! fis2 g1
  }
}
```



Véase también

Referencia de funcionamiento interno: el grabador Sección "Accidental_engraver" in *Referencia de Funcionamiento Interno* y el objeto Sección "AccidentalSuggestion" in *Referencia de Funcionamiento Interno*.

Ligaduras mensurales blancas

Las ligaduras mensurales blancas están contempladas de forma limitada.

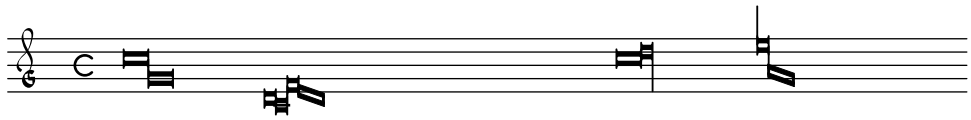
Para grabar ligaduras mensurales blancas, en el bloque layout o de diseño de página reemplace el grabador Sección "Ligature_bracket_engraver" in *Referencia de Funcionamiento Interno* por el grabador Sección "Mensural_ligature_engraver" in *Referencia de Funcionamiento Interno* en el contexto Sección "Voice" in *Referencia de Funcionamiento Interno*:

```
\layout {
  \context {
    \Voice
    \remove Ligature_bracket_engraver
    \consists Mensural_ligature_engraver
  }
}
```
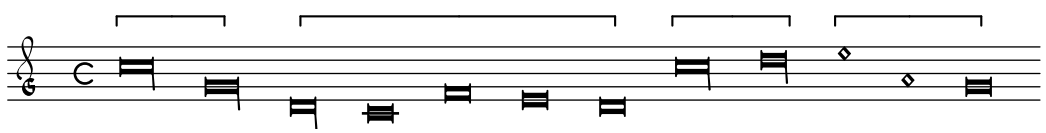
No existe ningún lenguaje de entrada adicional para describir la forma de una ligadura mensural blanca. Más bien la forma se determina exclusivamente a partir de la altura y duración de las notas comprendidas. Si bien este enfoque podría hacer que un usuario con poca experiencia tardase un tiempo en acostumbrarse a él, tiene la gran ventaja de que toda la información musical de la ligadura se conoce internamente. Esto no se requiere solamente para una salida MIDI correcta, sino que también permite la transcripción automática de las ligaduras.

Por ejemplo:

```
\score {
  \transpose c c' {
    \set Score.timing = ##f
    \set Score.defaultBarType = "empty"
    \override NoteHead #'style = #'neomensural
    \override Staff.TimeSignature #'style = #'neomensural
    \clef "petrucci-g"
    \[ c'\maxima g \]
    \[ d\longa c\breve f e d \]
    \[ c'\maxima d'\longa \]
    \[ e'1 a g\breve \]
  }
  \layout {
    \context {
      \Voice
      \remove Ligature_bracket_engraver
      \consists Mensural_ligature_engraver
    }
  }
}
```



Si no se sustituye el grabador Sección “Ligature_bracket_engraver” in *Referencia de Funcionamiento Interno* por el Sección “Mensural_ligature_engraver” in *Referencia de Funcionamiento Interno*, la misma música se transcribe de la siguiente manera:



Véase también

Advertencias y problemas conocidos

El espaciado horizontal de las ligaduras es pobre.

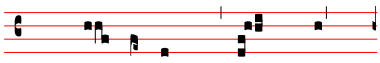
2.8.4 Tipografiado del canto gregoriano

Al componer tipográficamente una pieza en notación de canto gregoriano, el grabador Sección "Vaticana_ligature_engraver" in *Referencia de Funcionamiento Interno* selecciona automáticamente la cabeza adecuada de las figuras, de manera que no es neceario establecer explícitamente el estilo de las cabezas. A pesar de ello se puede establecer el estilo de las cabezas, p.ej.: a `vaticana_punctum` para producir neumas de punctum. De forma similar, el grabador Sección "Mensural_ligature_engraver" in *Referencia de Funcionamiento Interno* construye ligaduras mensurales automáticamente. Consulte [Ligaduras], página 285 para ver cómo funcionan los grabadores de ligaduras.

Contextos del canto gregoriano

El contexto predefinido de voz `VaticanaVoiceContext` y el de pauta `VaticanaStaffContext` se pueden utilizar para grabar una pieza de canto gregoriano en el estilo de la Editio Vaticana. Estos contextos establecen a unos valores iniciales adecuados todas las propiedades relevantes de los contextos y de los objetos gráficos, de forma que pueda inmediatamente comenzar a escribir el canto en sí, como lo demuestra el siguiente fragmento:

```
\include "gregorian.ly"
\score {
  <<
    \new VaticanaVoice = "cantus" {
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \]
      f \divisioMinima
      \[ f\melisma \pes a c' c' \pes d'\melismaEnd \]
      c' \divisioMinima \break
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \] f \divisioMinima
    }
    \new Lyrics \lyricsto "cantus" {
      San- ctus, San- ctus, San- ctus
    }
  >>
}
```



San- ctus, San- ctus,

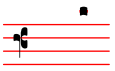
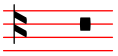
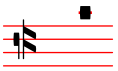


San- ctus

Véase también

Claves de canto gregoriano

La tabla siguiente muestra todas las claves gregorianas contempladas a través de la instrucción `\clef`. Algunas de las claves usan el mismo glifo, pero se diferencian solamente en la línea en que se imprimen. En estos casos, para enumerar estas claves se usa un número a continuación del nombre de la clave, contando desde la línea inferior. Aún se puede forzar manualmente la composición tipográfica de un glifo de clave sobre una línea arbitraria, como se describe en [Clave], página 12. La nota que se imprime a la derecha junto a cada una de las claves en la columna de los ejemplos, denota el Do central (`c'`) respecto a dicha clave.

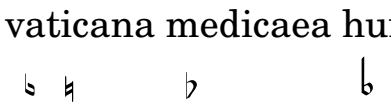
Descripción	**Claves posibles**	**Ejemplo**
Clave de Do, estilo Editio Vaticana	`vaticana-do1`, `vaticana-do2`, `vaticana-do3`	
Clave de Fa, estilo Editio Vaticana	`vaticana-fa1`, `vaticana-fa2`	
Clave de Do, estilo Editio Medicaea	`medicaea-do1`, `medicaea-do2`, `medicaea-do3`	
Clave de Fa, estilo Editio Medicaea	`medicaea-fa1`, `medicaea-fa2`	
Clave de Do, estilo hufnagel	`hufnagel-do1`, `hufnagel-do2`, `hufnagel-do3`	
Clave de Fa, estilo hufnagel	`hufnagel-fa1`, `hufnagel-fa2`	
Clave combinada de Do y Fa, estilo hufnagel	`hufnagel-do-fa`	

Véase también

Referencia de la notación: [Clave], página 12.

Alteraciones y armaduras de canto gregoriano

Están disponibles las alteraciones de los tres estilos gregorianos:



Como se ve en el ejemplo, cada estilo no contempla todas las alteraciones. Al intentar acceder a una alteración que no está contemplada, LilyPond cambia a un estilo diferente.

El estilo de las alteraciones y las armaduras de tonalidad se controla a través de la propiedad `glyph-name-alist` de los grobs (objetos gráficos) Sección "Accidental" in *Referencia de Funcionamiento Interno* y Sección "KeySignature" in *Referencia de Funcionamiento Interno*, respectivamente; p.ej.:

```
\override Staff.Accidental #'glyph-name-alist = #alteration-mensural-glyph-name-alist
```

Véase también

Referencia de la notación: Sección 1.1 [Alturas], página 1, [Alteraciones accidentales], página 4 y [Alteraciones accidentales automáticas], página 19 dan una introducción general del uso de las alteraciones. [Armadura de la tonalidad], página 15 ofrece una introducción general al uso de las armaduras de tonalidad.

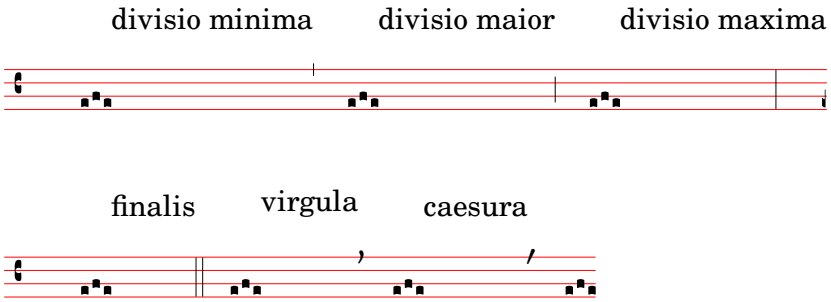
Referencia de funcionamiento interno: Sección "KeySignature" in *Referencia de Funcionamiento Interno*.

Divisiones

En la notación del canto gregoriano no existen silencios, sino [Divisiones], página 296.

Una *divisio* (en plural: *divisiones*, que en latín significa 'división') es un símbolo del contexto del pentagrama que se usa para estructurar la música gregoriana en frases y secciones. El significado musical de *divisio minima*, *divisio maior* y *divisio maxima* se podrían caracterizar como una pausa corta, media y larga (respectivamente), algo así como las marcas de respiración de [Marcas de respiración], página 94. El signo de *finalis* no sólo marca el final de un canto, sino que se usa también con frecuencia dentro de un único canto antifonal o responsorial para marcar el final de cada una de las secciones.

Para usar divisiones, incluya el archivo '`gregorian.ly`'. Contiene definiciones que podrá aplicar con tan sólo escribir `\divisioMinima`, `\divisioMaior`, `\divisioMaxima` y `\finalis` en los lugares adecuados de la entrada. Algunas ediciones usan *virgula* o *caesura* en vez de divisio minima. Por ello, '`gregorian.ly`' define también las instrucciones `\virgula` y `\caesura`.



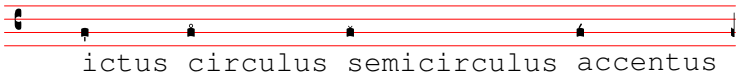
Instrucciones predefinidas

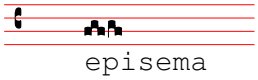
`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

Articulaciones del canto gregoriano

Además de los signos de articulación estándar que se hallan descritos en la sección [Articulaciones y ornamentos], página 84, se proveen signos de articulación diseñados específicamente para utilizarlos con la notación en el estilo de la *Editio Vaticana*.

```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \override TextScript #'font-family = #'typewriter
    \override TextScript #'font-shape = #'upright
    \override Script #'padding = #-0.1
    a\ictus_"ictus " \break
    a\circulus_"circulus " \break
    a\semicirculus_"semicirculus " \break
    a\accentus_"accentus " \break
    \[ a_"episema" \episemInitium \pes b \flexa a b \episemFinis \flexa a \]
  }
}
```





Véase también

Advertencias y problemas conocidos
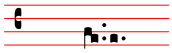
Ciertas articulaciones se colocan demasiado cerca verticalmente de las cabezas de nota correspondientes.

La línea episemática no se muestra en muchos casos. Si se presenta, el extremo derecho de la línea episemática se encuentra a menudo demasiado lejos hacia la derecha.

Puntos de aumentación (*morae*)

Los puntos de aumentación, llamados también *morae*, se escriben con la función musical `\augmentum`. Observe que `\augmentum` está programado como una función musical unaria y no como un prefijo de cabeza. Se aplica solamente a la expresión musical que sigue inmediatamente. Esto es, `\augmentum \virga c` no tendrá ningún efecto visible. En su lugar, escriba `\virga \augmentum c` o `\augmentum {\virga c}`. Tenga en cuenta también que puede escribir `\augmentum {a g}` como abreviatura de `\augmentum a \augmentum g`.

```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
    \augmentum g
  }
}
```



Véase también

Referencia de la notación: [Marcas de respiración], página 94.

Referencia de funcionamiento interno: Sección “BreathingSign” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Ancient notation” in *Fragmentos de código*.

Ligaduras de neumas cuadrados gregorianos

La notación de neumas cuadrados gregorianos está contemplada de forma limitada (siguiendo el estilo de la Editio Vaticana). El núcleo principal del conjunto de las ligaduras ya se pueden tipografiar, pero aún faltan aspectos esenciales para una composición tipográfica seria, como (entre otros) la alineación horizontal de varias ligaduras, la alineación de la letra y un correcto manejo de las alteraciones accidentales.

Se habilita el soporte de los neumas gregorianos por medio de la inclusión con `\include` del archivo `"gregorian.ly"` al principio del archivo. Esto hace que estén disponibles una cierta cantidad de instrucciones adicionales para producir los símbolos de los neumas que se usan en la notación de canto llano.

Las cabezas de las notas se pueden *modificar* y/o *unir*.

- La forma de la cabeza se puede modificar *precediendo* el nombre de la nota con una cualquiera de las instrucciones siguientes: `\virga`, `\stropha`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.
- Las ligaduras hablando en propiedad, (es decir, notas unidas), se producen escribiendo una de las instrucciones de unión `\pes` o `\flexa`, para el movimiento ascendente y descendente, respectivamente, *intercaladas* entre las notas que se han de unir.

Una nota sin modificaciones produce un *punctum*. Todos los demás neumas, incluso los neumas de una nota con forma distinta como la *virga*, se consideran en principio como ligaduras y por ello se deben escribir dentro de `\[...\]`.

Neumas de una nota:

- El *punctum* es la forma básica de nota (en el estilo *Vaticana*: un cuadrado con una ligera curvatura a modo de excelencia tipográfica). Además del *punctum* normal, están el *punctum inclinatum*, oblicuo, producido con el prefijo `\inclinatum`. El *punctum* normal se puede modificar con `\cavum`, que produce una nota vacía, y `\linea`, que traza líneas verticales a ambos lados de la nota.
- La *virga* tiene una plica descendente en el lado derecho. Se produce mediante el modificador `\virga`.

Ligaduras

A diferencia de casi todos los otros sistemas de notación de neumas, el aspecto tipográfico de las ligaduras no viene dictado directamente por las instrucciones de la entrada, sino que sigue unas convenciones que dependen del significado musical. Por ejemplo, una ligadura de tres notas con la forma musical bajo-alto-bajo, como `\[ a \pes b \flexa g \]`, produce un Torculus que consiste en tres cabezas de Punctum, mientras que la forma alto-bajo-alto, como `\[ a \flexa g \pes b \]`, produce un Porrectus con una forma de flexa curvada y una sola cabeza de Punctum. No existe ninguna instrucción para tipografiar explícitamente la forma de flexa curvada; la decisión de cuándo tipografiar una forma de flexa curvada está basada en la entrada musical. La idea de este enfoque es separar los aspectos musicales de la entrada, del estilo de notación de la salida. De esta forma, la misma entrada se puede reutilizar para tipografiar la misma música en un estilo diferente de notación de canto gregoriano.

Neumas licuescentes

Otra categoría fundamental de notas en el canto gregoriano es la de los llamados neumas licuescentes. Se utilizan bajo ciertas circunstancias al final de una sílaba que acaba en una letra 'licuescente', es decir, consonantes sonoras que pueden tener una altura tonal, (nasales, l, r, v, j, y sus diptongos equivalentes). Así, los neumas licuescentes nunca se utilizan aisladamente (aunque se pueden producir algunos de ellos), y siempre están al final de una ligadura.

Los neumas licuescentes se representan gráficamente de dos formas distintas más o menos intercambiables: con una nota pequeña o 'girando' la nota principal hacia arriba o hacia abajo. La primera forma se produce haciendo un `pes` o una `flexa` normales y modificando la forma de la segunda nota: `\[ a \pes \deminutum b \]` , mientras que la segunda se hace modificando la forma de un neuma de una nota con `\auctum` y uno de los generadores de dirección `\descendens` o `\ascendens`, p.ej. `\[ \auctum \descendens a \]` .

Símbolos especiales

Hay una tercera categoría de símbolos que se hace a partiru de un pequeño número de símbolos que tienen un significado especial (que, por cierto, casi siempre se conocen sólo vagamente): el *quilisma*, el *oriscus* y el *strophicus*. Todos ellos se producen anteponiendo al nombre de la nota el modificador correspondiente, `\quilisma`, `\oriscus` o `\stropha`.

Dentro de los delimitadores de ligadura `\[` y `\]`, se pueden acumular prácticamente cualquier cantidad de cabezas de nota para formar una sola ligadura, y los prefijos de cabeza como `\pes`, `\flexa`, `\virga`, `\inclinatum`, etc. se pueden mezclar libremente. El uso del conjunto de reglas que subyace a la construcción de ligaduras en la tabla anterior está consecuentemente extrapolada. De esta manera se pueden crear un número infinito de ligaduras distintas.

Observe que el uso de esos símbolos en la propia música sigue ciertas reglas que LilyPond no comprueba. P.ej., el *quilisma* siempre es la nota intermedia de una ligadura ascendente, y suele caer sobre un intervalo de semitono, pero es perfectamente posible, si bien incorrecto, hacer un quilisma de una nota.

Además de los símbolos de nota, el archivo gregorian.ly define también las instrucciones `\versus`, `\responsum`, `\ij`, `\iij`, `\IJ` y `\IIJ`, que producen los caracteres correspondientes, p.ej. para utilizarlos en la letra, como marcas de sección, etc. Estas instrucciones utilizan caracteres de Unicode especiales y sólo funcionan si se usa una fuente tipográfica que los contemple.

La tabla siguiente muestra un conjunto limitado, pero representativo, de ligaduras gregorianas, junto a los fragmentos de código que las producen. La tabla está basada en la tabla de neumas extendidos del segundo volumen del Antiphonale Romanum (*Liber Hymnarius*), publicado en 1983 por los monjes de Solesmes. La primera columna da el nombre de la ligadura, con la forma principal en tipo negrita y las formas licuescentes en cursiva. La tercera forma muestra el fragmento de código que produce dicha ligadura, utilizando Sol, La y Si como alturas de ejemplo.

Neumas de una nota

Formas **Básica** y *Licuescente*	**Salida**	**Código de Lilypond**
Punctum		`\[ b \]`

```
\[ \cavum b \]
```



```
\[ \linea b \]
```


Punctum Auctum Ascendens

```
\[ \auctum \ascendens b \]
```



Punctum Auctum Descendens

```
\[ \auctum \descendens b \]
```



Punctum inclinatum

```
\[ \inclinatum b \]
```



Punctum Inclinatum Auctum

```
\[ \inclinatum \auctum b \]
```



Punctum Inclinatum Parvum

```
\[ \inclinatum \deminutum b \]
```


Virga



Ligaduras de dos notas

Clivis vel Flexa

```
\[ b \flexa g \]
```



Clivis Aucta Descendens

```
\[ b \flexa \auctum \descendens
g \]
```



Clivis Aucta Ascendens

```
\[ b \flexa \auctum \ascendens
g \]
```



Cephalicus

```
\[ b \flexa \deminutum g \]
```



Podatus o Pes

```
\[ g \pes b \]
```



Pes Auctus Descendens

```
\[ g \pes \auctum \descendens b
\]
```



Pes Auctus Ascendens

```
\[ g \pes \auctum \ascendens b
\]
```



Epiphonus

```
\[ g \pes \deminutum b \]
```



Pes Initio Debilis

```
\[ \deminutum g \pes b \]
```



Pes Auctus Descendens Initio Debilis

```
\[ \deminutum g \pes \auctum
\descendens b \]
```



Ligaduras de varias notas

Torculus

```
\[ a \pes b \flexa g \]
```



Torculus Auctus Descendens

```
\[ a \pes b \flexa \auctum
\descendens g \]
```



Torculus Deminutus

```
\[ a \pes b \flexa \deminutum g
\]
```


Torculus Initio Debilis

```
\[ \deminutum a \pes b \flexa g
\]
```



Torculus Auctus Descendens Initio Debilis

```
\[ \deminutum a \pes b \flexa
\auctum \descendens g \]
```



Torculus Deminutus Initio Debilis

```
\[ \deminutum a \pes b \flexa
\deminutum g \]
```



Porrectus

```
\[ a \flexa g \pes b \]
```



Porrectus Auctus Descendens

```
\[ a \flexa g \pes \auctum
\descendens b \]
```



Porrectus Deminutus

```
\[ a \flexa g \pes \deminutum b
\]
```



Climacus

```
\[ \virga b \inclinatum a
\inclinatum g \]
```



Climacus Auctus

```
\[ \virga b \inclinatum a
\inclinatum \auctum g \]
```



Climacus Deminutus

```
\[ \virga b \inclinatum a
\inclinatum \deminutum g \]
```



Scandicus

```
\[ g \pes a \virga b \]
```



Scandicus Auctus Descendens

```
\[ g \pes a \pes \auctum
\descendens b \]
```



Scandicus Deminutus

```
\[ g \pes a \pes \deminutum b \]
```



Símbolos especiales

Quilisma

```
\[ g \pes \quilisma a \pes b \]
```



Quilisma Pes Auctus Descendens

```
\[ \quilisma g \pes \auctum
\descendens b \]
```



Oriscus

```
\[ \oriscus b \]
```


Pes Quassus

```
\[ \oriscus g \pes \virga b \]
```



Pes Quassus Auctus Descendens

```
\[ \oriscus g \pes \auctum
\descendens b \]
```



Salicus

```
\[ g \oriscus a \pes \virga b \]
```



Salicus Auctus Descendens

```
\[ g \oriscus a \pes \auctum
\descendens b \]
```



(Apo)stropha

```
\[ \stropha b \]
```



Stropha Aucta

```
\[ \stropha \auctum b \]
```



Bistropha

```
\[ \stropha b \stropha b \]
```



Tristropha

```
\[ \stropha b \stropha b
\stropha b \]
```



Trigonus

```
\[ \stropha b \stropha b
\stropha a \]
```



Instrucciones predefinidas

Están contemplados los siguientes prefijos de cabeza: `\virga`, `\stropha`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.

Se pueden acumular los prefijos de cabeza, aunque con ciertas restricciones. Por ejemplo, se pueden aplicar `\descendens` o `\ascendens` a una cabeza, pero no los dos a la misma cabeza.

Se pueden ligar dos cabezas adyacentes con las instrucciones infijas `\pes` y `\flexa` para una línea melódica ascendente y descendente, respectivamente.

Utilice la función musical unaria `\augmentum` para añadir puntos de augmentum.

Véase también

Advertencias y problemas conocidos

Cuando aparece un punto de `\augmentum` al final de la última pauta dentro de una ligadura, a veces se posiciona defectuosamente en el sentido vertical. Para rodear el problema, escriba una nota adicionall de salto (p.ej. `s8`) como última nota de la pauta.

`\augmentum` debería estar implementado como un prefijo de cabeza en vez de una función musical unaria, de forma que `\augmentum` se pudiera entremezclar con los prefijos de cabeza en un orden arbitrario.

2.8.5 Trabajar con música antigua: escenarios y soluciones

El trabajo con música antigua implica con frecuencia tareas adicionales que difieren considerablemente de la notación moderna para la que se diseñó Lilypond. En el resto de esta sección se perfilan unos cuantos escenarios típicos, con sugerencias de soluciones. Entre ellos están los siguientes:

- cómo hacer incipits (o sea, material a modo de prefacio que indica cómo era el original) a las transcripciones modernas de música mensural;
- cómo cosneguir el diseño *Mensurstriche* que se usa con frecuencia para las transcripciones modernas de música polifónica;
- cómo transcribir canto gregoriano en notación moderna;
- cómo generar notación antigua y moderna a partir de la misma fuente.

Incipits

Continuará...

Véase también

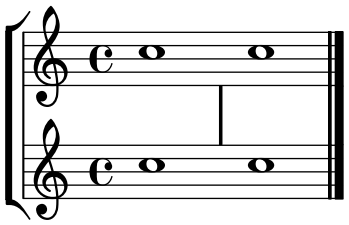
Disposición tipo «Mensurstriche»

Mensurstriche (‘líneas de mensuración’) es el término aceptado para las barras de compás que se trazan entre los pentagramas de un sistema pero no dentro de los propios pentagramas. Es una forma común de preservar el aspecto visual de las duraciones del original, es decir, sin tener que dividir las notas sincopadas en las barras de compás, y al mismo tiempo ofrecer la ayuda orientativa que proporcionan las barras de compás.

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir con un `StaffGroup` en vez de un `ChoirStaff`. La línea divisoria sobre los pentagramas se borra estableciendo la propiedad `transparent`.

```
global = {
  \override Staff.BarLine #'transparent = ##t
  s1 s
  % la barra final no se interrumpe
  \revert Staff.BarLine #'transparent
  \bar "|."
}
\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
```

```
  >>
}
```



Véase también

Transcripción de canto gregoriano

El canto gregoriano se puede transcribir a notación moderna con ciertos trucos sencillos.

Plicas. Todas las plicas se pueden eliminar de una vez retirando el grabador `Stem_engraver` del contexto de voz:

```
\layout {
  ...
  \context {
    \Voice
      \remove "Stem_engraver"
  }
}
```

Sin embargo, en ciertos estilos de transcripción, se usan plicas de forma ocasional, por ejemplo para indicar la transición a partir de un recitativo monotónico a un gesto melódico fijo. En estos casos podemos usar bien `\override Stem #'transparent = ##t` o bien `\override Stem #'length = #0`, y restaurar la plica cuando se necesite con el correspondiente `\once \override Stem #'transparent = ##f` (véase el ejemplo de abajo).

Compás. Para el canto no medido, existen diversas alternativas.

Se puede eliminar el grabador Time_signature_engraver del contexto de Staff sin ningún efecto secundario negativo. La alternativa, que es hacer transparente la indicación de compás, deja un espacio en blanco en la partitura, pues la indicación invisible sigue ocupando un espacio.

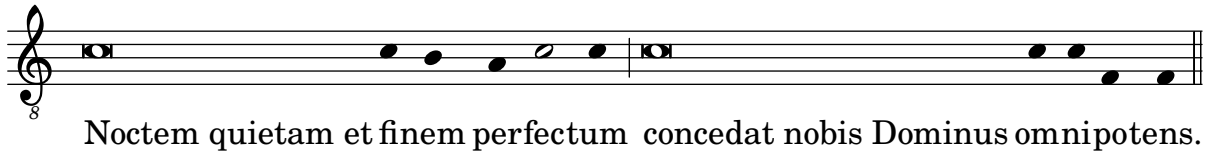
En muchos casos da un buen resultado `\set Score.timing = ##f`. Otra alternativa es utilizar `\\CadenzaOn` y `\CadenzaOff`.

Para quitar las barras de compás, el enfoque radical consiste en retirar (mediante la instrucción `\remove`) el grabador Bar_engraver del contexto de pentagrama Staff. Una vez más, en vez de ello podemos usar `\override BarLine #'transparent = ##t` si se necesita ocasionalmente una barra de compás.

Un tipo de transcripción muy común es el canto de recitativo, en el que la nota repetida se indica con una breve. Se puede tratar el texto del tono de recitativo de dos maneras. Podemos ponerlo como una sola sílaba alineada por la izquierda:

```
\include "gregorian.ly"
canto = \relative c' {
  \clef "G_8"
  c\breve c4 b4 a c2 c4  \divisioMaior
  c\breve c4 c f, f \finalis
}
```

```
verba = \lyricmode {
  \once \override LyricText #'self-alignment-X = #-1
  "Noctem quietam et" fi -- nem per -- fec -- tum
  \once \override LyricText #'self-alignment-X = #-1
  "concedat nobis Dominus" om -- ni -- po -- tens.
}
\score {
  \new Staff <<
  \new Voice = "melodia" \canto
  \new Lyrics = "one" \lyricsto melodia \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \override Stem #'transparent = ##t
    }
  }
}
```



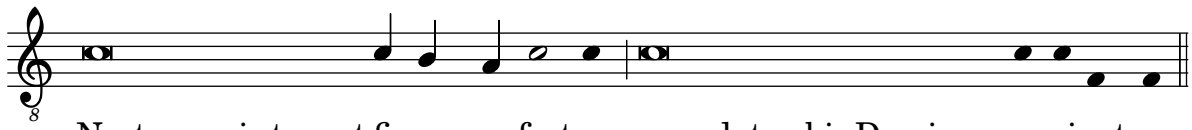
Esto funciona bien en la medida en que el texto no comprenda un salto de línea. Si es el caso, una alternativa sería añadir notas ocultas a la partitura, en combinación con cambios en la visibilidad de las plicas:

```
\include "gregorian.ly"
canto = \relative c' {
  \clef "G_8"
  \set Score.timing = ##f
  c\breve \override NoteHead #'transparent = ##t  c c c c c
  \revert NoteHead #'transparent
  \override Stem #'transparent = ##f \stemUp c4 b4 a
  \override Stem #'transparent = ##t c2 c4  \divisioMaior
  c\breve \override NoteHead #'transparent = ##t c c c c c c c
  \revert NoteHead #'transparent c4 c f, f \finalis
}

verba = \lyricmode {
  No -- ctem qui -- e -- tam et fi -- nem per -- fec -- tum
  con -- ce -- dat no -- bis Do -- mi -- nus om -- ni -- po -- tens.
}

\score {
  \new Staff <<
    \new Voice = "melodia" \canto
    \new Lyrics \lyricsto "melodia" \verba
```

```
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \override BarLine #'transparent = ##t
      \override Stem #'transparent = ##t
    }
  }
}
```



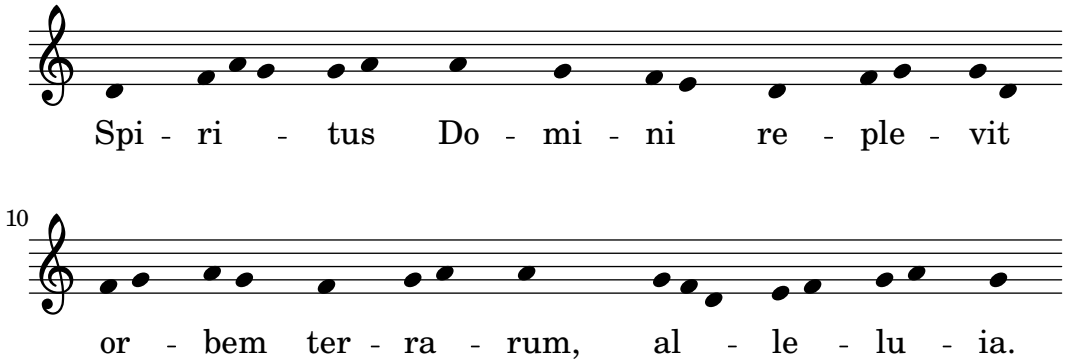
Otra situación común es la transcripción de cantos neumáticos o melismáticos, es decir, cantos con un número variable de notas en cada sílaba. En este caso, podemos establecer los grupos de sílabas claramente separados, normalmente también las subdivisiones de un melisma mayor. Una forma de conseguirlo es usar un `\time` fijo, p.ej. 1/4, y dejar que cada sílaba o grupo de notas llene uno de esos compases, con la ayuda de grupos de valoración especial o duraciones más breves. Si las líneas divisorias y todas las demás indicaciones rítmicas se hacen transparentes, y el espacio que rodea las líneas divisorias se aumenta, ello proporciona una aceptable representación del original, en notación moderna.

Para evitar que las sílabas de distinta anchura (como "-ri" y "-rum") dispongan los grupos de notas de las sílabas de forma irregular, se puede ajustar la propiedad `#'X-extent` del objeto `LyricText` a un valor fijo. Otra forma más engorrosa sería añadir las sílabas como elementos de `\markup`. Si se necesitan más ajustes posteriormente, esto se puede hacer fácilmente con 'notas' `s`.

```
spiritus = \relative c' {
  \time 1/4
  \override Lyrics.LyricText #'X-extent  = #'(0 . 3)
  d4 \times 2/3 { f8 a g } g a a4 g f8 e
  d4 f8 g g8 d f g a g f4 g8 a a4  s
  \times 2/3 { g8 f d } e f g a g4
}

letraSpir = \lyricmode {
  Spi -- ri -- _ _ tus  _ Do -- mi -- ni  _ re -- ple -- _ vit _
  or -- _ bem _  ter -- ra -- _ rum, al -- _ _ le -- _ lu
  -- _ ia.
}
\score {
  \new Staff <<
    \new Voice = "canto" \spiritus
    \new Lyrics = "one" \lyricsto "canto" \letraSpir
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
```

```
      \override BarLine #'X-extent = #'(-1 . 1)
      \override Stem #'transparent = ##t
      \override Beam #'transparent = ##t
      \override BarLine #'transparent = ##t
      \override TupletNumber #'transparent = ##t
    }
  }
}
```



Véase también

Antigua y moderna desde la misma fuente

Continuará...

Véase también

Anotaciones editoriales

Continuará...

Véase también

2.9 Músicas del mundo

El propósito de esta sección es remarcar los problemas de notación que son de relevancia a las tradiciones musicales distintas a la occidental.

2.9.1 Música árabe

Esta sección destaca asuntos relativos a la música árabe.

Referencias para música árabe

La música árabe ha sido principalmente una tradición oral hasta el momento. Cuando se transcribe la música, se suele hacer en la forma de un esbozo sobre el que los intérpretes deben improvisar ampliamente. En una medida creciente, se está adoptando la notación occidental con ciertas variantes para preservar y transmitir la música árabe.

Algunos elementos de la notación musical occidental como la transcripción de acordes o las particellas independientes, no se necesitan para tipografiar las piezas árabes, más tradicionales.

Sin embargo hay ciertos problemas, como la necesidad de indicar intervalos que tienen una medida intermedia entre el semitono y el tono, además de los intervalos mayores y menores de la música occidental. También existe la necesidad de agrupar e indicar los numerosos maqams (modos) que forman parte de la música árabe.

Por lo general la notación musical árabe no trata de indicar con precisión los elementos microtonales que se encuentran presentes en la práctica musical.

Son de relevancia para la música árabe ciertos asuntos que se encuentran cubiertos en otras partes del manual:

- Los nombres de las notas y sus alteraciones (entre ellos, cuartos de tono) se pueden realizar como se estudia en [Nombres de las notas en otros idiomas], página 6.
- Las armaduras adicionales también pueden realizarse como se describe en [Armadura de la tonalidad], página 15.
- Los compases complejos pueden requerir agrupaciones manuales de las notas tal y como se describe en [Barras manuales], página 67.
- Los *takasim* o improvisaciones libres desde el punto de vista rítmico se pueden escribir omitiendo las barras de compás como aparece en [Música sin compasear], página 49.

Véase también

Referencia de la notación: [Nombres de las notas en otros idiomas], página 6, [Armadura de la tonalidad], página 15, [Barras manuales], página 67.
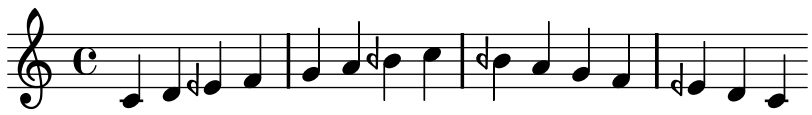
Fragmentos de código: Sección “World music” in *Fragmentos de código*.

Nombres de nota árabes

Los nombres árabes tradicionales de las notas pueden llegar a ser bastante largos y no son adecuados para la escritura musical, y por ello no se utilizan. Los nombres ingleses de las notas no son muy familiares dentro de la educación musical árabe, por lo que en su lugar se utilizan los nombres italo-franco-españoles o del solfeo (`do, re, mi, fa, sol, la, si`). También se pueden utilizar los modificadores (alteraciones), como se estudia en [Nombres de las notas en otros idiomas], página 6.
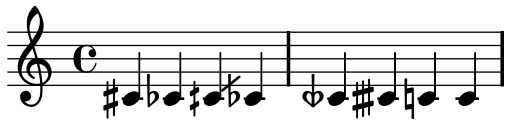
Por ejemmplo, así es como puede ser la notación de la escala árabe *rast*:

```
\include "arabic.ly"
\relative do' {
  do re misb fa sol la sisb do sisb la sol fa misb re do
}
```



El símbolo del semi-bemol no coincide con el símbolo que se utiliza en la notación árabe. Se puede usar el símbolo `\dwn` definido en `arabic.ly` precediendo a un bemol como vía alternativa si es importante usar el símbolo árabe específico del semi-bemol. El aspecto del síimbolo del semi-bemol dentro de la armadura no se puede alterar mediante este método.

```
\include "arabic.ly"
\relative do' {
  \set Staff.extraNatural = ##f
  dod dob dosd \dwn dob dobsb dodsd do do
}
```



Véase también

Referencia de la notación: [Nombres de las notas en otros idiomas], página 6.

Fragmentos de código: Sección “World music” in *Fragmentos de código*.

Armaduras de tonalidad árabes

Además de las armaduras de tonalidad mayores y menores, están definidas en `arabic.ly` las siguientes tonalidades: *bayati*, *rast*, *sikah*, *iraq* y *kurd*. Estas armaduras de tonalidad definen una pequeña cantidad de grupos de maqam, en contraste con los numerosos maqams que son de uso común.

En general, un maqam usa la armadura de su grupo o la de un grupo vecino, y las alteraciones que difieren se van marcando a lo largo de la música.

Por ejemplo, para indicar la armadura de una pieza en el maqam muhayer:

```
\key re \bayati
```

Aquí, *re* es la altura predeterminada del maqam muhayer, y *bayati* es el nombre del maqam básico dentro del grupo.

Al tiempo que la armadura indica el grupo, el título suele indicar el maqam específico, y así en este ejemplo el nombra del maqam muhayer debe aparecer en el título.

Se pueden indicar de la misma forma otros maqams dentro del mismo grupo bayati, como aparece en la tabla a continuación: bayati, hussaini, saba y ushaq. Todos ellos son variantes del maqam básico y más común dentro del grupo, que es bayati. Normalmente se diferencian del maqam básico en su tectacordo superior, o en ciertos detalles de discurso que no cambian su naturaleza fundamental como parientes.

El otro maqam del grupo (Nawa) se relaciona con el bayati a través de una modulación que se indica en la tabla entre paréntesis para aquellos maqams que son modulaciones de su maqam básico. Los maqams árabes admiten sólo modulaciones limitadas, a causa de la naturaleza de los instrumentos musicales árabes. El Nawa se puede indicar de la siguiente forma:

```
\key sol \bayati
```

En la música árabe, el mismo término que se utiliza para indicar un grupo de maqam, como bayati, es también un maqam que normalmente es el más importante del grupo, y podemos verlo como el maqam básico.

A continuación presentamos una posibilidad de agrupación que asigna los maqams más comunes a armaduras de tonalidad:

grupo de maqam	**tono**	**final**	**otros maqams del grupo (final)**
ajam	major	sib	jaharka (fa)
bayati	bayati	re	hussaini, muhayer, saba, ushaq, nawa (sol)
hijaz	kurd	re	shahnaz, shad arban (sol), hijazkar (do)
iraq	iraq	sisb	-
kurd	kurd	re	hijazkar kurd (do)
nahawand	minor	do	busalik (re), farah faza (sol)
nakriz	minor	do	nawa athar, hisar (re)
rast	rast	do	mahur, yakah (sol)
sikah	sikah	misb	huzam

Fragmentos de código seleccionados

Armaduras de tonalidad no tradicionales

La muy utilizada instrucción `\key` establece la propiedad `keySignature` property, dentro del contexto `Staff`.
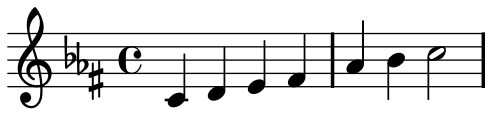
Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

`\set Staff.keySignature = #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)` donde, para cada elemento dentro de la lista, `octava` especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), `paso` especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y `alteración` es `,SHARP ,FLAT ,DOUBLE-SHARP` etc. (observe la coma precedente.)

Alternativamente, para cada elemento de la lista el uso del formato más conciso `(paso . alteración)` especifica que la misma alteración debe estar en todas las octavas.

He aquí un ejemplo de una posible armadura para generar una escala exátona:

```
\relative c' {
  \set Staff.keySignature = #`(((0 . 3) . ,SHARP)
                              ((0 . 5) . ,FLAT)
                              ((0 . 6) . ,FLAT))
  c4 d e fis
  aes4 bes c2
}
```



Véase también

Referencia de la notación: [Armadura de la tonalidad], página 15.

Manual de aprendizaje: Sección “Accidentals and key signatures” in *Manual de Aprendizaje*.

Referencia de funcionamiento interno: Sección “KeySignature” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “World music” in *Fragmentos de código*, Sección “Pitches” in *Fragmentos de código*.

Indicaciones de compás árabes

Ciertas formas clásicas de música árabe y turca como el *Semai* utilizan compases poco comunes como el 10/8. Ello puede derivar en una agrupación automática de las notas muy distinta a la de las partituras impresas existentes, donde las notas pueden no estar agrupadas sobre el pulso, sino de una forma difícil de emular ajustando el barrado automático. Este comportamiento se puede sobreescribir desactivando el barrado automático e indicándolo manualmente. Cuando no es problema la coincidencia con partituras impresas existentes, aún podemos ajustar el comportamiento del barrado y/o utilizar compases compuestos.

Fragmentos de código seleccionados
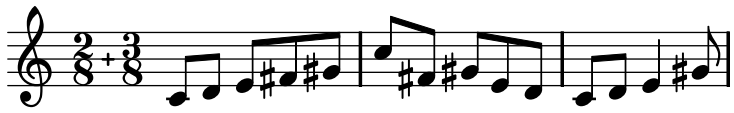
Indicaciones de compases compuestos

Las indicaciones de compás poco frecuentes como `"5/8"` se pueden ejecutar como compases compuestos (p.ej. `"3/8 + 2/8"`), que combinan dos o más metros diferentes. LilyPond puede hacer la música de este tipo fácil de leer e interpretar, imprimiendo explícitamente las indicaciones

de compás compuesto y adaptando el comportamiento automático de las barras (también se pueden añadir indicaciones gráficas de la agrupación de compases; véase el fragmento de código apropiado en la base de datos).

```
#(define ((compound-time one two num) grob)
  (grob-interpret-markup grob
    (markup #:override '(baseline-skip . 0) #:number
      (#:line (
          (#:column (one num))
          #:vcenter "+"
          (#:column (two num))))
      )))

\relative c' {
  \override Staff.TimeSignature #'stencil = #(compound-time "2" "3" "8")
  \time 5/8
  #(override-auto-beam-setting '(end 1 8 5 8) 1 4)
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```

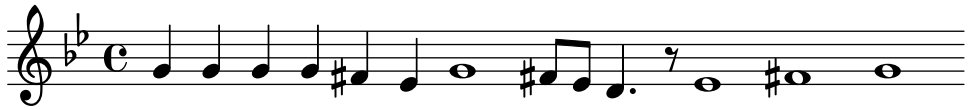


Improvisación de música árabe

Para las improvisaciones o *taqasim* que son libres durante unos momentos, se puede omitir la indicación de compás y se puede usar `\cadenzaOn`. Podría ser necesario ajustar el estilo de alteraciones accidentales, porque la ausencia de líneas divisorias hará que la alteración aparezca una sola vez. He aquí un ejemplo de cómo podría ser el comienzo de una improvisación *hijaz*:

```
\include "arabic.ly"

\relative sol' {
  \key re \kurd
  #(set-accidental-style 'forget)
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
```



Véase también

Referencia de la notación: [Barras manuales], página 67, [Barras automáticas], página 56, [Música sin compasear], página 49, [Alteraciones accidentales automáticas], página 19, [Establecer el comportamiento de las barras automáticas], página 58, [Indicación de compás], página 46.
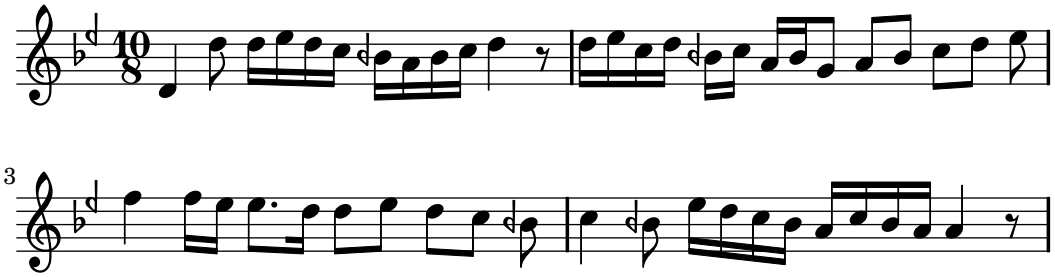
Fragmentos de código: Sección “World music” in *Fragmentos de código*.

Ejemplo de música árabe

A continuación aparece una plantilla que emplea también el comienzo de un Semai turco muy común en la educación musical árabe con el objeto de ilustrar algunas de las peculiaridades de la notación árabe, como los intervalos intermedios y los modos poco usuales que tratamos en esta sección.

```
\include "arabic.ly"
\score {
  \relative re' {
    \set Staff.extraNatural = ##f
    \set Staff.autoBeaming = ##f
    \key re \bayati
    \time 10/8

    re4 re'8 re16 [misb re do] sisb [la sisb do] re4 r8
    re16 [misb do re] sisb [do] la [sisb sol8] la [sisb] do [re] misb
    fa4 fa16 [misb] misb8. [re16] re8 [misb] re  [do] sisb
    do4 sisb8 misb16 [re do sisb] la [do sisb la] la4 r8
  }
  \header {
    title = "Semai Muhayer"
    composer = "Jamil Bek"
  }
}
```



Véase también

Fragmentos de código: Sección “World music” in *Fragmentos de código*

Lecturas adicionales

1. The music of the Arabs, de Habib Hassan Touma [Amadeus Press, 1996], contiene un tratado de los maqams y sus formas de agrupación.

   También existen varios sitios web que explican los maqams, y algunos de ellos ofrecen ejemplos de audio, como:

   - http://www.maqamworld.com/
   - http://www.turath.org/

   Existen ciertas variantes en los detalles sobre la forma en que se agrupan los maqams, a pesar del relativo acuerdo existente acerca de los criterios de agrupación de maqams en función de los tetracordos inferiores comunes, o por medio de la modulación.

2. No es completamente consistente, incluso a veces dentro del mismo texto, la forma en que se deben especificar las armaduras para los maqams concretos. Sin embargo, es usual utilizar una armadura por grupo, más que una armadura distinta para cada maqam concreto.

Varios métodos de los siguientes autores contienen ejemplos de composiciones turcas y árabes, principalmente.

- Charbel Rouhana
- George Farah
- Ibrahim Ali Darwish Al-masri

3 Entrada y salida generales

Esta sección trata de asuntos relacionados con la entrada y salida generales de LilyPond, más que con la notación específica.

3.1 Estructura del código de entrada

El formato principal de entrada para LilyPond son archivos de texto. Por convenio, el nombre de estos archivos termina en `.ly`.

3.1.1 Estructura de una partitura

Ub bloque `\score` debe contener una expresión musical única delimitada por un par de llaves:
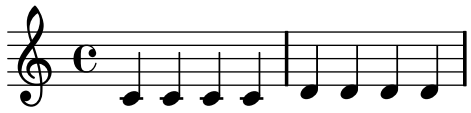
```
\score {
...
}
```

> **Nota:** Sólo puede haber **una** expresión musical dentro del bloque `\score`, y **debe** estar encerrada entre llaves.

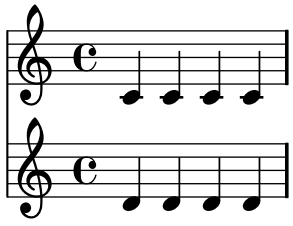
Esta expresión musical única puede tener cualquier tamaño, y puede contener otras expresiones musicales hasta cualquier grado de complejidad. Todos los siguientes ejemplos son expresiones musicales:

```
{ c'4 c' c' c' }
```

```
{
  { c'4 c' c' c'}
  { d'4 d' d' d'}
}
```



```
<<
  \new Staff { c'4 c' c' c' }
  \new Staff { d'4 d' d' d' }
>>
```



```
{
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { \flute }
      \new Staff { \oboe }
    >>
    \new StaffGroup <<
      \new Staff { \violinI }
```

```
      \new Staff { \violinII }
    >>
  >>
}
```

Los comentarios son una excepción a esta regla general (para ver otras excepciones, consulte Sección 3.1.3 [Estructura del archivo], página 318). Tanto los comentarios de una línea como los que están delimitados por `%{ .. %}` se pueden escribir en cualquier lugar dentro del archivo de entrada. Se pueden escribir dentro o fuera del bloque `\score`, y dentro o fuera de la expresión musical única dentro del bloque `\score`.

Véase también

Manual de aprendizaje: Sección “Working on input files” in *Manual de Aprendizaje*, Sección “Music expressions explained” in *Manual de Aprendizaje*, Sección “Score is a (single) compound musical expression” in *Manual de Aprendizaje*.

3.1.2 Varias partituras en un libro

Un documento puede contener varias piezas de música y texto. Son ejemplos un libro de estudios o una parte orquestal con varios movimientos. Cada movimiento se introduce con un bloque `\score`:

```
\score {
  ..música..
}
```

y los textos se introducen con un bloque `\markup`:

```
\markup {
  ..texto..
}
```

Todos los movimientos y textos que aparecen en el mismo archivo `.ly` se tipografían normalmente en la forma de un solo archivo de salida.

```
\score {
  ..
}
\markup {
  ..
}
\score {
  ..
}
```

Sin embargo, si queremos varios archivos de salida a partir del mismo archivo `.ly`, podemos escribir varios bloques `\book`, donde cada uno de estos bloques `\book` dará como resultado un archivo de salida distinto. Si no especificamos ningún bloque `\book` en el archivo,, LilyPond trata implícitamente todo el archivo como un solo bloque `\book`, véase Sección 3.1.3 [Estructura del archivo], página 318. Una excepción importante se da dentro de los documentos de lilypondbook, en los que tenemos que escribir explícitamente un bloque `\book`, pues en caso contrario sólo aparecerá en la salida el primer `\score` o `\markup`.

El encabezamiento de cada pieza se puede poner dentro del bloque `\score`. Antes de cada movimiento, se imprime el nombre de la pieza (`piece`) extraído del encabezamiento. El título del libro entero se puede poner dentro del bloque `\book`, pero si no está presente se inserta el encabezamiento `\header` que está al principio del archivo.

```
\header {
  title = "Ocho miniaturas"
```

```
  composer = "Igor Stravinsky"
}
\score {
  ...
  \header { piece = "Romanze" }
}
\markup {
   ..texto de la segunda estrofa..
}
\markup {
   ..texto de la tercera estrofa..
}
\score {
  ...
  \header { piece = "Menuetto" }
}
```

Se pueden agrupar distintas piezas de música en partes de un libro utilizando bloques `\bookpart`. Las partes de libro se separan por medio de saltos de página, y pueden dar comienzo con un título, como el libro propiamente dicho, mediante la especificación del bloque `\header`.

```
\bookpart {
  \header {
    title = "Título del libro"
    subtitle = "Primera parte"
  }
  \score { ... }
  ...
}
\bookpart {
  \header {
    subtitle = "Segunda parte"
  }
  \score { ... }
  ...
}
```

3.1.3 Estructura del archivo

Un archivo `.ly` puede contener cualquier número de expresions del nivel superior, donde una expresión del nivel superior es una de las siguientes posibilidades:

- Una definición de salida, como `\paper`, `\midi` y `\layout`. Tal definición en el nivel más alto cambia los ajustes predeterminados para todo el libro. Si se introduce más de una de estas definiciones del mismo tipo en el nivel más alto, cualquier definición en una expresión posterior tiene prioridad.
- Una expresión directa de Scheme, como `#(set-default-paper-size "a7" 'landscape)` o `#(ly:set-option 'point-and-click #f)`.
- Un bloque `\header`. Esto establece el bloque de encabezamiento global. Es el bloque que contiene las definiciones de los ajustes para todo el libro, como el compositor, título, etc.
- Un bloque `\score`. Esta partitura se reúne con otras partituras del nivel superior, y se combina con ellas formando un `\book` (libro) único. Este comportamiento se puede cambiar estableciendo el valor de la variable `toplevel-score-handler` en el nivel superior. El *handler* predeterminado se define en el archivo de inicio '`../scm/lily.scm`'.

- Un bloque `\book` combina lógicamente varios movimientos (es decir, varios bloques `\score`) en un solo documento. Si hay varias partituras `\score`, se crea un solo archivo de salida para cada bloque `\book`, en el que se concatenan todos los movimientos correspondientes. La únca razón para especificar explícitamente bloques `\book` en un archivo de entrada `.ly` es si queremos crear varios archivos de salida a partir de un único archivo de entrada. Una excepción se da dentro de los documentos de lilypond-book, en los que tenemos que escribir explícitamente un bloque `\book` si queremos tener más de una única partitura `\score` o marcado `\markup` en el mismo ejemplo. Se puede cmbiar este comportamiento estableciendo el valor de la variable `toplevel-book-handler` en el nivel superior. El *handler* predeterminado se define en el archivo de inicio '`../scm/lily.scm`'.
- Un bloque `\bookpart`. Un libro puede dividirse en varias partes, utilizando bloques `\bookpart`, con objeto de facilitar la inserción de saltos de página o usar distintos ajustes de `\paper` en las distintas partes.
- Una expresión musical compuesta, como

  ```
  { c'4 d' e'2 }
  ```

  Esto añade la pieza al `\score` y la conforma dentro de un libro único junto a todas las demás partituras `\score` y expresiones musicales del nivel superior. En otras palabras, un archivo que contiene sólo la expresión musical anterior se traduce a

  ```
  \book {
    \score {
      \new Staff {
        \new Voice {
          { c'4 d' e'2 }
        }
      }
    }
  \layout { }
  \header { }
  }
  ```

  Se puede cambiar este comportamiento estableciendo el valor de la bariable `toplevel-music-handler` en el nivel más alto. El *handler* predeterminado se define en el archivo de inicio '`../scm/lily.scm`'.
- Un texto de marcado, por ejemplo una estrofa

  ```
  \markup {
     2.  Estrofa dos, línea uno
  }
  ```

  Los textos de marcado se colocan encima, en medio o debajo de las partituras o expresiones musicales, según donde aparezcan escritos.
- Una variable, como

  ```
  fulano = { c4 d e d }
  ```

  Esto se puede utilizar más tarde en el archivo escribiendo `\fulano`. El nombre de las variables ha de ser exclusivamente alfabético; sin números, guiones ni barras bajas.

El ejemplo siguiente muestra tres cosas que se pueden introducir en el nivel superior:

```
\layout {
  % No justificar la salida
  ragged-right = ##t
}

\header {
```

```
   title = "Do-re-mi"
}

{ c'4 d' e2 }
```

En cualquier punto del archivo se pueden introducir cualquiera de las instrucciones léxicas siguientes:

- `\version`
- `\include`
- `\sourcefilename`
- `\sourcefileline`
- Un comentario de una línea, determinado por un símbolo `%` al principio.
- Un comentario de varias líneas, delimitado por `%{ .. %}`.

Véase también

Manual de aprendizaje: Sección "How LilyPond input files work" in *Manual de Aprendizaje*.

3.2 Títulos y cabeceras

Casi toda la música impresa tiene un título y el nombre del compositor; ciertas piezas tienen mucha más información.

3.2.1 Crear títulos

Se crean títulos para cada uno de los bloques `\score` así como para todo el archivo de entrada (o bloque `\book`, libro) y las partes de libro (creadas mediante bloques `\bookpart`).

El contenido de cada título se extrae a partir de los bloques `\header` (cabecera). El bloque de cabecera de un libro contempla lo siguiente:

`dedication`
: El dedicatario de la música, centrado en lo alto de la primera página.

`title`
: El título de la música, centrado justo debajo de la dedicatoria.

`subtitle`
: Subtítulo, centrado debajo del título.

`subsubtitle`
: Sub-subtítulo, centrado debajo del subtítulo.

`poet`
: Nombre del poeta, a la izquierda, debajo del subtítulo.

`composer`
: Nombre del compositor, a la derecha, debajo del subtítulo.

`meter`
: Texto de la medida, a la izquierda, debajo del poeta.

`opus`
: Nombre del Opus, a la derecha, debajo del compositor.

`arranger`
: Nombre del arreglista, a la derecha, debajo del Opus.

`instrument`
: Nombre del instrumento, centrado, debajo del arreglista. También aparece centrado en lo alto de cada página (aparte de la primera).

`piece`
: Nombre de la pieza, a la izquierda, debajo del instrumento.

`breakbefore`
: Esto fuerza que el título comience sobre una página nueva (establézcalo a ##t o a ##f).

`copyright`
: Aviso de Copyright, centrado en la parte inferior de la primera página. Para insertar el símbolo de copyright symbol, consulte Sección 3.3.3 [Codificación del texto], página 332.

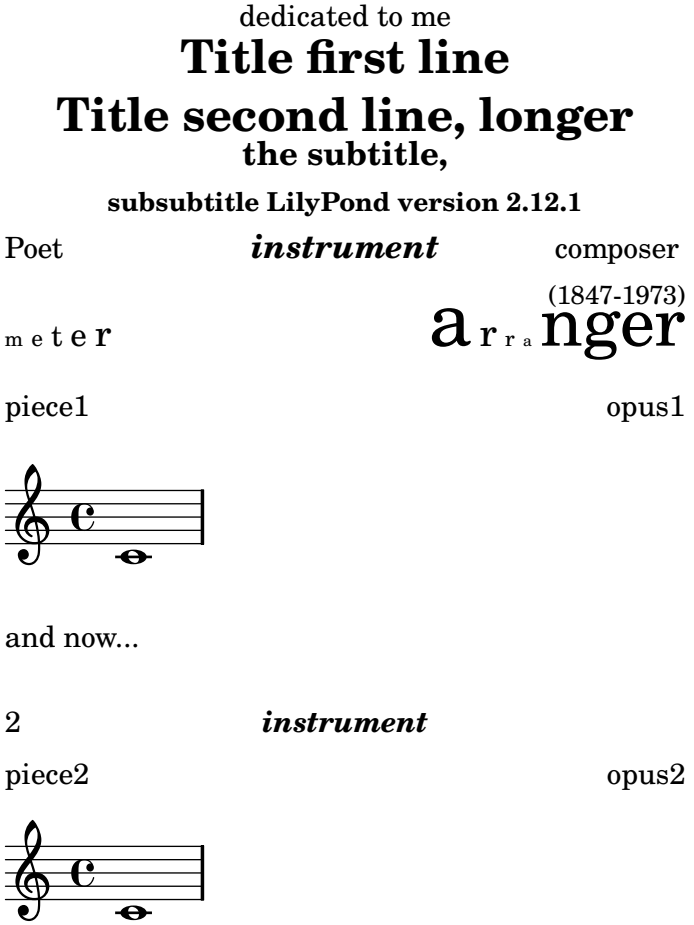
`tagline` Cartel de propósito general en el pie de página, centrado al final de la última.

A continuación presentamos una demostración de todos los campos que están disponibles. Fíjese en que puede utilizar en la cabecera cualquier instrucción de marcado de texto (Véase Sección 1.8.2 [Formatear el texto], página 173).

```
\paper {
  line-width = 9.0\cm
  paper-height = 10.0\cm
}

\book {
  \header {
    dedication = "dedicated to me"
    title = \markup \center-column { "Title first line" "Title second line,
longer" }
    subtitle = "the subtitle,"
    subsubtitle = #(string-append "subsubtitle LilyPond version "
(lilypond-version))
    poet = "Poet"
    composer =  \markup \center-column { "composer" \small "(1847-1973)" }
    texttranslator = "Text Translator"
    meter = \markup { \teeny "m" \tiny "e" \normalsize "t" \large "e" \huge
"r" }
    arranger = \markup { \fontsize #8.5 "a" \fontsize #2.5 "r" \fontsize
#-2.5 "r" \fontsize #-5.3 "a" \fontsize #7.5 "nger" }
    instrument = \markup \bold \italic "instrument"
    piece = "Piece"
  }

  \score {
    { c'1 }
    \header {
      piece = "piece1"
      opus = "opus1"
    }
  }
  \markup {
      and now...
  }
  \score {
    { c'1 }
    \header {
      piece = "piece2"
      opus = "opus2"
    }
  }
}
```



Music engraving by LilyPond 2.12.1—www.lilypond.org

Como se mostró anteriormente, puede usar varios bloques `\header`. Cuando los mismos campos aparecen en distintos bloques, tiene validez el último en definirse. A continuación un breve ejemplo.

```
\header {
  composer = "Compositor"
}
\header {
  piece = "Pieza"
}
\score {
  \new Staff { c'4 }
  \header {
    piece = "Otra pieza"  % sobreescribe a la anterior
```

```
  }
}
```

Si define la cabecera `\header` dentro del bloque `\score`, por lo general se imprimirán solamente las cabeceras `piece` y `opus`. Fíjese en que la expresión musical debe aparecer antes del `\header`.

```
\score {
  { c'4 }
  \header {
    title = "title"  % no impreso
    piece = "piece"
    opus = "opus"
  }
}
```

piece opus



Se puede cambiar este comportamiento (e imprimir todas las cabeceras cuando la cabecera `\header` está definida dentro del bloque `\score`) mediante la utilización de

```
\paper{
  print-all-headers = ##t
}
```

El pie de página por omisión se encuentra vacío, excepto para la primera página, en la que se inserta el campo de `copyright` extraído del `\header`, y para la última página, en la que se escribe el cartel `tagline` sacado del bloque `\header`. La línea por omisión para este cartel es "Music engraving by LilyPond (*version*)".[1]

Las cabeceras se pueden quitar completamente estableciendo su valor a falso.

```
\header {
  tagline = ##f
  composer = ##f
}
```

3.2.2 Títulos personalizados

Una opción más avanzada es cambiar las definiciones de las variables siguientes dentro del bloque `\paper` block. El archivo de inicio '`ly/titling-init.ly`' ofrece un listado de la disposición predeterminada.

`bookTitleMarkup`
: Es el título que se escribe en lo alto del documento de salida completo. Normalmente contiene el compositor y el título de la pieza.

`scoreTitleMarkup`
: Es el título que se coloca por encima de un bloque `\score`. normalmente contiene el nombre del movimiento (campo `piece`).

`oddHeaderMarkup`
: Es el encabezamiento de las páginas de numeración impar.

[1] Las partituras bien tipografiadas son una magnífica promoción para nosotros, así que por favor, si puede, conserve intacta la línea del letrero.

`evenHeaderMarkup`
: Es el encabezamiento de las páginas de numeración par. Si se deja sin especificar, se usará el encabezamiento de las páginas impares.

  De forma predeterminada, los encabezamientos se definen de tal forma que el número de página está en el borde exterior, y el isntrumento está centrado.

`oddFooterMarkup`
: Es el pie de las páginas de numeración impar.

`evenFooterMarkup`
: Es el pie de las páginas de numeración par. Si se deja sin especificar, se usará en su lugar el encabezamiento de las páginas impares.

  De forma predeterminada, el pie de página tiene el aviso de copyright en la primera página, y el cartel `tagline` en la última.

La definición siguiente pone el título en el extremo izquierdo, y el compositor en el extremo derecho sobre una única línea.

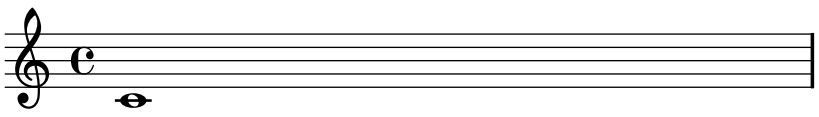
```
\paper {
  bookTitleMarkup = \markup {
   \fill-line {
     \fromproperty #'header:title
     \fromproperty #'header:composer
   }
  }
}
```

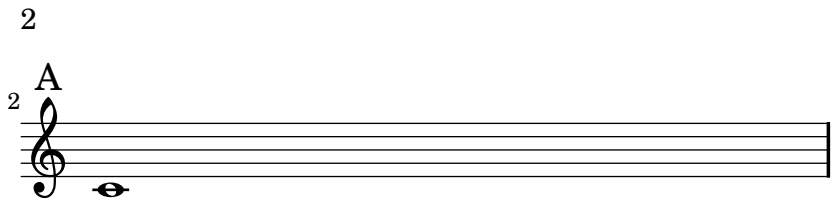
3.2.3 Referencias a números de página

Se puede marcar un lugar determinado de una partitura utilizando la instrucción `\label` (etiqueta), bien en lo alto de la estructura o bien dentro de la música. Posteriormente se puede hacer referencia a esta etiqueta dentro de un elemento de marcado, para obtener el número de la página en que se encuentra la marca, usando la instrucción de marcado `\page-ref`.

```
\header { tagline = ##f }
\book {
  \label #'firstScore
  \score {
    {
      c'1
      \pageBreak \mark A \label #'markA
      c'
    }
  }

  \markup { The first score begins on page \page-ref #'firstScore "0" "?" }
  \markup { Mark A is on page \page-ref #'markA "0" "?" }
}
```





The first score begins on page 1

Mark A is on page 2

La instrucción de marcado `\page-ref` toma tres argumentos:

1. la etiqueta, un símbolo de scheme, p.ej. `#'firstScore`;
2. un elemento de marcado que se usará como medidor para estimar las dimensiones del marcado;
3. un elemento de marcado que se utilizará en sustitución del número de página si la etiqueta es desconocida.

El motivo de que se necesite un medidor es que en el momento en que se están interpretando los marcados, los saltos de página aún no se han producido y por tanto los números de página no se conocen todavía. Para sortear este inconveniente, la interpretación real del marcado se retrasa hasta un momento posterior; sin embargo, las dimensiones del marcado se tienen que conocer de antemano, así que se usa el medidor para decidir estas dimensiones. Si el libro tiene entre 10 y 99 páginas, el medidor puede ser `"00"`, es decir, un número de dos dígitos.

Instrucciones predefinidas

`\label`, `\page-ref`.

3.2.4 Índice general

Se puede insertar un índice general o tabla de contenidos utilizando la instrucción `\markuplines \table-of-contents`. Los elementos que deben aparecer en la tabla de contenidos se introducen con la instrucción `\tocItem`, que se puede usar en el nivel más alto de la jerarquía del código, o dentro de una expresión musical.

```
\markuplines \table-of-contents
\pageBreak

\tocItem \markup "Primera partitura"
\score {
  {
    c'  % ...
    \tocItem \markup "Un punto concreto dentro de la primera partitura"
    d'  % ...
  }
}

\tocItem \markup "Segunda partitura"
\score {
  {
    e' % ...
  }
}
```

Los elementos de marcado que se usan para dar formato al índice general se encuentran definidos dentro del bloque `\paper`. Los elementos predeterminados son `tocTitleMarkup`, para dar formato al título de la tabla, y `tocItemMarkup`, para aplicar formato a los elementos del

índice, que consisten en el título del elemento y el número de página. Estas variables se pueden cambiar por parte del usuario:

```
\paper {
  %% Traducir el título del índice al español:
  tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Índice general" \null }
    \hspace #1
  }
  %% usar una fuente mayor
  tocItemMarkup = \markup \large \fill-line {
    \fromproperty #'toc:text \fromproperty #'toc:page
  }
}
```

Observe la forma en que nos referimos al texto y al número de página del elemento de índice, dentro de la definición `tocItemMarkup`.

También se pueden definir nuevas instrucciones y elementos de marcado para crear índices generales más eleborados:

- en primer lugar, defina una variable de marcado nueva dentro del bloque `\paper`
- luego defina una función de música cuyo propósito es añadir un elemento al índice general utilizando esta variable de marcado del bloque `\paper`.

En el ejemplo siguiente se define un estilo nuevo para introducir los nombres de los actos de una ópera en el índice general:

```
\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}

tocAct =
#(define-music-function (parser location text) (markup?)
   (add-toc-item! 'tocActMarkup text))
```

Table of Contents

Atto Primo

Coro. Viva il nostro Alcide 1
Cesare. Presti omai l'Egizzia terra 1

Atto Secondo

Sinfonia 1
Cleopatra. V'adoro, pupille, saette d'Amore 1

Véase también

Archivos de inicio: ‘`ly/toc-init.ly`’.

Instrucciones predefinidas

`\table-of-contents`, `\tocItem`.

3.3 Trabajar sobre los archivos de entrada

3.3.1 Inclusión de archivos de LilyPond

Un proyecto grande se puede dividir en varios archivos. Para referirse a otro archivo, utilice

```
\include "otroarchivo.ly"
```

La línea `\include "otroarchivo.ly"` equivale a pegar todo el contenido de ‘`otroarchivo.ly`’ en el archivo actual en el lugar en que aparece el `\include`. Por ejemplo, en un proyecto grande podríamos querer archivos distintos para cada parte instrumental y crear un archivo de “partitura completa” que reúne los archivos individuales de los instrumentos. Normalmente el archivo incluido define un cierto número de variables que a partir de entonces quedan disponibles para poderlas utilizar en el archivo de la partitura completa. En los archivos incluidos se pueden marcar las secciones etiquetadas para ayudar a hacerlas utilizables en distintos lugares de la partitura, véase Sección 3.3.2 [Distintas ediciones a partir de una misma fuente], página 328.

Los archivos que están en el directorio de trabajo actual se pueden referenciar simplemente especificando el nombre depués de la instrucción `\include`. Los archivos en otros lugares se pueden incluir proporcionando una referencia de ruta completa o una ruta relativa (pero utilice la barra inclinada normal del UNIX, /, no la barra invertida de DOS/Windows, \, como separador de directorio). Por ejemplo, si ‘`material.ly`’ está situado un directorio por encima del directorio de trabajo actual, utilice

```
\include "../material.ly"
```

o si los archivos de las partes orquestales incluidas están todos situados en un subdirectorio llamado ‘`partes`’ dentro del directorio actual, use

```
\include "partes/VI.ly"
\include "partes/VII.ly"
... etc.
```

Los archivos de inclusión también pueden contener enunciados `\include` a su vez. Estos enunciados `\include` de segundo nivel no se interpretan hasta que se incorporan al archivo principal, por lo que los nombres de archivo que especifican deben ser relativos al directorio que contiene el archivo principal, no al directorio que contiene el archivo incluido.

También se pueden incluir archivos de un directorio que está en una ruta de búsqueda especificada como opción al invocar a LilyPond desde la línea de órdenes. Los archivos incluidos se especifican usando solamente su nombre de archivo. Por ejemplo, para compilar mediante este método un archivo ‘`principal.ly`’ que incluye archivos situados en un subdirectorio llamado ‘`partes`’, cambie al directorio que contiene ‘`principal.ly`’ e introduzca

```
lilypond --include=partes principal.ly
```

y en ‘`principal.ly`’ escriba

```
\include "VI.ly"
\include "VII.ly"
... etc.
```

Los archivos que se incluyen en muchas partituras se pueden poner en el directorio de LilyPond ‘`../ly`’ (la localización de este directorio depende de la instalación: véase Sección “Other

sources of information" in *Manual de Aprendizaje*). Estos archivos se pueden incluir sencillamente por su nombre en un enunciado `\include`. De esta forma se incluyen los archivos dependientes del idioma como '`english.ly`'.

LilyPond incluye de forma predeterminada ciertos archivos cuando se inicia el programa. Estas inclusiones no son visibles para el usuario, pero los archivos se pueden identificar ejecutando `lilypond --verbose` desde la línea de órdenes. Esto presentará una lista de rutas y archivos que utiliza LilyPond, junto a muchas otras indicaciones. De forma alternativa, pueden verse explicaciones sobre los archivos más importantes de este conjunto en Sección "Other sources of information" in *Manual de Aprendizaje*. Estos archivos se pueden editar, pero los cambios realizados sobre ellos se perderán al instalar una nueva versión de LilyPond.

Pueden verse ejemplos sencillos sobre el uso de `\include` en Sección "Scores and parts" in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección "Other sources of information" in *Manual de Aprendizaje*, Sección "Scores and parts" in *Manual de Aprendizaje*.

Advertencias y problemas conocidos

Si un archivo incluido recibe un nombre igual al de uno de los archivos de instalación de LilyPond, el archivo que tiene prioridad es el de los archivos de instalación de LilyPond.

3.3.2 Distintas ediciones a partir de una misma fuente
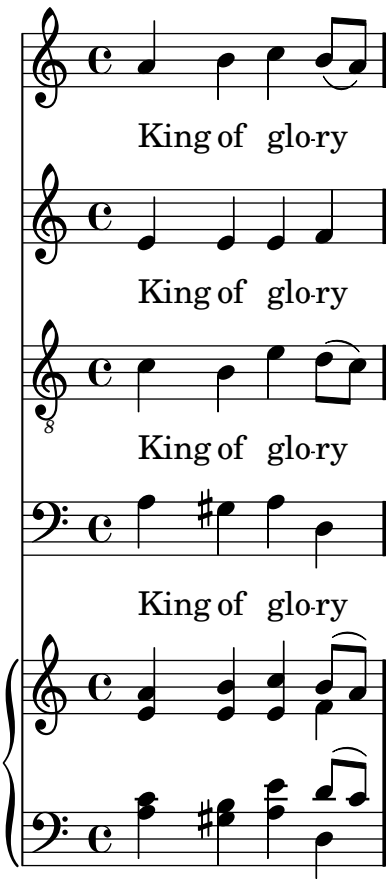
Se encuentran disponibles varios mecanismos para facilitar la producción de distintas versiones de una partitura a partir de la misma fuente musical. Las variables son quizá más útiles para combinar secciones extensas de música y/o anotaciones de diversas formas, mientras que las etiquetas son más útiles para seleccionar una de entra varias secciones de música más cortas. Cualquiera que sea el método utilizado, la separación de la notación de la estructura de la partitura hará que sea más fácil cambiar la estructura dejando intacta la notación.

Uso de las variables

Si las secciones de la música se definen dentro de variables, se pueden reutilizar en distintas partes de la partitura, véase Sección "Organizing pieces with variables" in *Manual de Aprendizaje*. Por ejemplo, una partitura vocal *a cappella* con frecuencia incluye una reducción de piano de las partes para ensayar que ees idéntica a la música vocal, por lo que ésta sólo se tiene que escribir una vez. Se puede combinar sobre un solo pentagrama la música de dos variables, véase [Combinación automática de las partes], página 120. He aquí un ejemplo:

```
musicaSoprano = \relative c'' { a4 b c b8( a)}
musicaContralto = \relative g' { e4 e e f }
musicaTenor = \relative c' { c4 b e d8( c) }
musicaBajo = \relative c' { a4 gis a d, }
letraCompleta = \lyricmode {King of glo -- ry }
<<
  \new Staff = "Soprano" \musicaSoprano
  \new Lyrics \letraCompleta
  \new Staff = "Alto" \musicaContralto
  \new Lyrics \letraCompleta
  \new Staff = "Tenor" {
    \clef "treble_8"
    \musicaTenor
  }
  \new Lyrics \letraCompleta
```

```
  \new Staff = "Bajo" {
    \clef "bass"
    \musicaBajo
  }
  \new Lyrics \letraCompleta
  \new PianoStaff <<
    \new Staff = "MD" {
      \set Staff.printPartCombineTexts = ##f
      \partcombine
      \musicaSoprano
      \musicaContralto
    }
    \new Staff = "MI" {
      \set Staff.printPartCombineTexts = ##f
      \clef "bass"
      \partcombine
      \musicaTenor
      \musicaBajo
    }
  >>
>>
```



Se pueden producir partituras distintas que presenten sólo las partes vocales o sólo la parte de piano, cambiando solamente los enunciados estructurales, sin modificar la notación musical.

Para partituras extensas, las definiciones de variable se pueden poner en archivos separados que se incluyen más tarde, véase Sección 3.3.1 [Inclusión de archivos de LilyPond], página 327.

Uso de etiquetas

La instrucción `\tag #'parteA` marca una expresión musical con el nombre *parteA*. Las expresiones etoiquetadas de esta manera se pueden seleccionar o filtrar más tarde por su nombre, usando bien `\keepWithTag #'nombre` o bien `\removeWithTag #'nombre`. El resultado de la aplicación de estos filtros a la música etiquetada es como sigue:

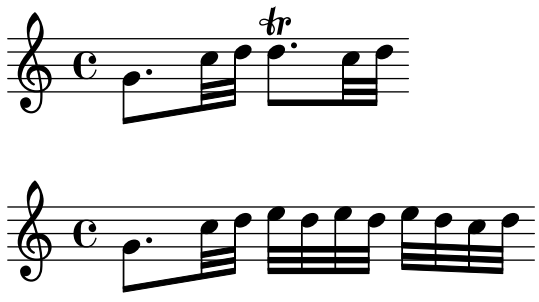
Filtro	**Resultado**
Música etiquetada precedida de `\keepWithTag #'nombre`	Se incluye la música no etiquetada y la música etiquetada con *nombre*; se excluye la música etiquetada con cualquier otro nombre de etiqueta.
Música etiquetada precedida de `\removeWithTag #'nombre`	Se incluye la música no etiquetada y la música etiquetada con cualquier nombre de etiqueta distinto a *nombre*; se excluye la música etiquetada con *nombre*.
Música etiquetada no precedida de `\keepWithTag` ni de `\removeWithTag`	Se incluye toda la música etiquetada y no etiquetada.

Los argumentos de las instrucciones `\tag`, `\keepWithTag` y `\removeWithTag` deben ser un símbolo (como `#'partitura` o `#'parte`), seguido de una expresión musical.

En el ejemplo siguiente vemos dos versiones de una pieza musical, una que muestra trinos con la notación usual y otra con los trinos expandidos explícitamente:

```
musica = \relative g' {
  g8. c32 d
  \tag #'trills {d8.\trill }
  \tag #'expand {\repeat unfold 3 {e32 d} }
  c32 d
 }

\score {
  \keepWithTag #'trills \musica
}
\score {
  \keepWithTag #'expand \musica
}
```

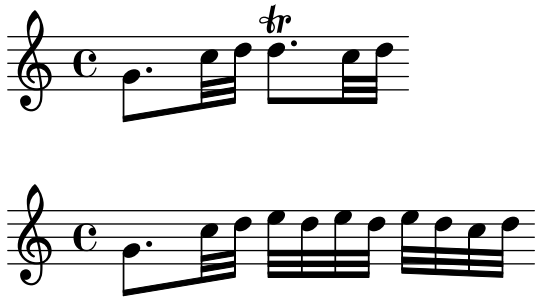


De forma alternativa, a veces es más fácil excluir secciones de música:

```
musica = \relative g' {
  g8. c32 d
  \tag #'trills {d8.\trill }
  \tag #'expand {\repeat unfold 3 {e32 d} }
  c32 d
```

```
 }

\score {
  \removeWithTag #'expand
  \musica
}
\score {
  \removeWithTag #'trills
  \musica
}
```



El flitrado con etiquetas se puede aplicar a articulaciones, textos, etc. anteponiendo

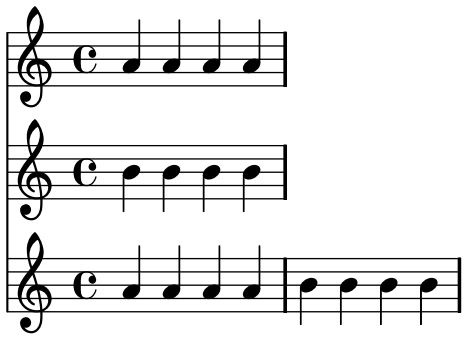
`-\tag #'`*`mi-etiqueta`*

a una articulación. Por ejemplo, esto define una nota con una indicación opcional de digitación y una nota con una anotación condicional:

```
c1-\tag #'finger ^4
c1-\tag #'warn ^"¡Ojo!"
```
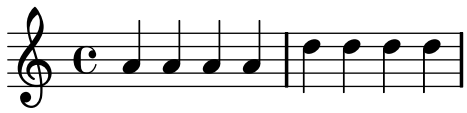
Se pueden poner varias etiquetas sobre expresiones con varias entradas de `\tag`:

```
musica = \relative c'' {
  \tag #'a \tag #'both { a a a a }
  \tag #'b \tag #'both { b b b b }
}
<<
\keepWithTag #'a \musica
\keepWithTag #'b \musica
\keepWithTag #'both \musica
>>
```



Se pueden aplicar varios filtros `\removeWithTag` a una sola expresión musical para quitar varias secciones etiquetadas con nombres distintos:

```
musica = \relative c'' {
\tag #'A { a a a a }
\tag #'B { b b b b }
\tag #'C { c c c c }
\tag #'D { d d d d }
}
{
\removeWithTag #'B
\removeWithTag #'C
\musica
}
```



Dos o más filtros `\keepWithTag` aplicados a una sola expresión musical producen la eliminación de *todas* las secciones etiquetadas, porque el primer filtro quita todas las secciones etiquetadas excepto la que se nombra, y el segundo filtro elimina incluso esa sección etiquetada.

Véase también

Manual de aprendizaje: Sección “Organizing pieces with variables” in *Manual de Aprendizaje*.

Referencia de la notación: [Combinación automática de las partes], página 120, Sección 3.3.1 [Inclusión de archivos de LilyPond], página 327.

3.3.3 Codificación del texto

LilyPond usa el conjunto de caracteres definido por el consorcio Unicode y la norma ISO/IEC 10646. Define un nombre único y un código para los conjuntos de caracteres que se utilizan en prácticamente todos los idiomas modernos y también en muchos otros. Unicode se puede implementar utilizando varios esquemas de codificación distintos. LilyPond usa la codificación UTF-8 (UTF son las siglas de *Unicode Transformation Format*, o formato de transformación de Unicode) que representa todos los caracteres comunes de Latin en un solo byte, y representa otros caracteres usando un formato de longitud variable de hasta cuatro bytes.
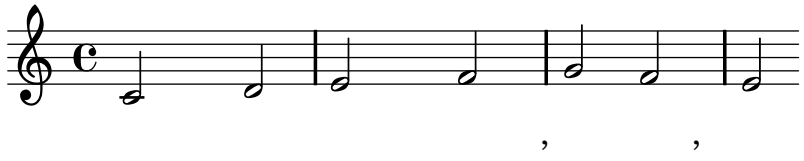
El aspecto visual real de los caracteres viene determinado por los glifos que se definen en las fuentes tipográficas concretas que se tengan disponibles: una fuente tipográfica define la asignación de un subconjunto de los códigos de Unicode a glifos. LilyPond usa la biblioteca Pango para representar y disponer tipográficamente textos multilingües.

Lilypond no realiza ninguna conversión en la codificación de la entrada. Esto significa que cualquier text, ya sea el título, la letra de la canción o una instrucción musical que contenga caracteres distintos a los del conjunto ASCII, se deben codificar en UTF-8. La forma más fácil de escribir dicho texto es utilizar un editor preparado para Unicode y guardar el archivo con la codificación UTF-8. Casi todos los editores modernos populares contemplan el UTF-8, por ejemplo lo hacen vim, Emacs, jEdit, y GEdit. Todos los sistemas MS Windows posteriores a NT usan Unicode como codificación de caracteres nativa, de manera que incluso el accesorio Bloc de Notas (Notepad) puede editar y guardar un archivo en el formato UTF-8. Una alternativa más funcional para Windows es BabelPad.

Si un archivo de entrada que contiene un carácter que no es ASCII, no se guarda en el formato UTF-8, se genera el mensaje de error

```
FT_Get_Glyph_Name () error: argumento inválido
```

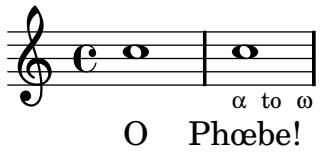
He aquí un ejemplo que muestra texto cirílico, hebreo y portugués:



Para introducir un caracter aislado para el que se conoce la secuencia de escape de Unicode pero no está disponible en el editor que se está usando, use `\char ##xhhhh` dentro de un bloque `\markup`, donde `hhhh` es el código hexadecimal del carácter en cuestión. Por ejemplo, `\char ##x03BE` inserta el carácter Unicode U+03BE, que tiene el nombre Unicode "Letra griega Xi pequeña". Se puede sustituir cualquier código hexadecimal Unicode, y si todos los caracteres especiales se escriben en este formato no es necesario guardar el archivo de entrada en formato UTF-8. Por supuesto, debe estar instalada y estar disponible para Lilypond una fuente tipográfica que contenga codificados todos estos caracteres.

El ejemplo siguiente muestra caracteres codificados en UTF-8 que se usan en cuatro lugares: en una llamada de ensayo, como texto de articulación, en la letra y como texto independiente bajo la partitura:

```
\score {
  \relative c'' {
    c1 \mark \markup { \char ##x03EE }
    c1_\markup { \tiny { \char ##x03B1 " to " \char ##x03C9 } }
  }
  \addlyrics { O \markup { \concat{ Ph \char ##x0153 be! } } }
}
\markup { "Copyright 2008" \char ##x00A9 }
```



Copyright 2008 ©

Para escribir el símbolo de copyright en la nota de créditos, utilice:

```
\header {
  copyright = \markup { \char ##x00A9 "2008" }
}
```

3.3.4 Presentar la notación de LilyPond

La impresión textual de una expresión musical en notación de LilyPond puede hacerse usando la función musical `\displayLilyMusic`. Por ejemplo,

```
{
  \displayLilyMusic \transpose c a, { c e g a bes }
}
```

imprimirá

```
{ a, cis e fis g }
```

De forma predeterminada, LilyPond imprime estos mensajes en la consola junto al resto de los mensajes. Para discernir entre estos mensajes y guardar el resultado de `\display{MATERIAL}`, redireccione la salida hacia un archivo.

```
lilypond archivo.ly >resultado.txt
```

3.4 Controlar la salida

3.4.1 Extracción de fragmentos de música

Es posible citar pequeños fragmentos de una partitura grande directamente a partir de la salida. Puede compararse con la separación de una pieza de una partitura, recortándola con tijeras.

Esto se hace definiendo los compases que se deben recortar. Por ejemplo, la inclusión de la siguiente definición

```
\layout {
  clip-regions
  = #(list
      (cons
       (make-rhythmic-location 5 1 2)
       (make-rhythmic-location 7 3 4)))
}
```

extrae un fragmento empezando por la mitad del quinto compás y finalizando en el séptimo compás. El significado de `5 1 2` es: después de una blanca (1/2) en el compás 5, y el de `7 3 4` después de tres negras del compás 7.

Se pueden definir más zonas de recorte añadiendo más parejas de definición de tiempo rhythmic-location a la lista.

Para poder utilizar esta funcionalidad, LilyPond se debe invocar con la opción `-dclip-systems`. Los recortes aparecen en la salida como archivos EPS, y se convierten en PDF y PNG si estos formatos también están activados.

Para ver más información sobre los formatos de salida, consulte Sección “Invocar a LilyPond” in *Utilización del Programa*.

3.4.2 Saltar la música corregida

Al escribir o copiar música, normalmente sólo es interesante de ver y corregir la música cercana al final (donde estams añadiendo las notas). Para acelerar este proceso de corrección, es posible saltar la composición tipográfica de todos excepto unos pocos de los últimos compases. Esto se consigue poniendo

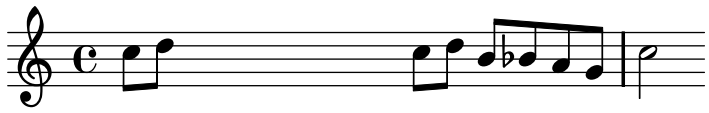
```
showLastLength = R1*5
\score { ... }
```

en nuestro archivo fuente. Esto hará que se tracen sólo los últimos cinco compases (suponniendo un compás de 4/4) de cada una de las partituras `\score` del archivo de entrada. Para piezas largas, el tipografiado de únicamente una parte pequeña es con frecuencia un orden de magnitud más rápido que el de la obra completa. Si estamos trabajando sobre el principio de una partitura que ya hemos tipografiado (p.ej., para añadir una parte instrumental nueva), también puede ser útil la propiedad `showFirstLength`.

Este mecanismo de pasar por alto partes de un partitura se puede controlar con un grano más fino mediante la propiedad `Score.skipTypesetting`. Cuando su valor está establecido, no se lleva a cabo ningún tipografiado en absoluto.

Esta propiedad se usa también para controlar la salida al archivo MIDI. Observe que pasa por alto todos los eventos, incluidos el tempo y los cambios de instrumento. Está usted advertido.

```
\relative c'' {
  c8 d
  \set Score.skipTypesetting = ##t
  e e e e e e e
  \set Score.skipTypesetting = ##f
  c d b bes a g c2 }
```



En música polifónica, `Score.skipTypesetting` afecta a todas las voces y pentagramas, ahorrando más tiempo incluso.

3.5 Salida MIDI

El MIDI (Musical Instrument Digital Interface, Interfase Digital para Instrumentos Musicales) es un estándar para interconectar y controlar instrumentos musicales electrónicos. Un archivo o secuencia MIDI es una serie de notas dentro de un conjunto de pistas. No es un archivoo de sonidos reales; se necesita un programa reproductor especial para traducir la serie de notas en sonidos de verdad.

Cualquier música se puede convertir a archivos MIDI, de manera que podamos escuchar lo que hayamos introducido. Esto es muy conveniente para comprobar la corrección de la música; las octavas equivocadas o las alteraciones erróneas se ponen de relieve muy claramente al escuchar la salida MIDI.

La salida MIDI reserva un canal por cada pentagrama, y otro para los ajustes globales. Por ello, el archivo MIDI no debe tener más de 15 pautas (o 14 si no se usa la percusión). Otros pentagramas permanecen en silencio.

3.5.1 Crear archivos MIDI

Para crear un MIDI a partir de una pieza de música, escriba un bloque `\midi` en la partitura, por ejemplo:

```
\score {
  ...música...
  \midi { }
}
```

Si hay un bloque `\midi` dentro de un `\score` que no tenga ningún bloque `\layout`, solamente se producirá MIDI. Si se necesita una notación impresa, debe haber también un bloque `\layout`:

```
\score {
  ...música...
  \midi { }
  \layout { }
}
```

Se traducen correctamente a la salida MIDI la altura y duración de las notas, las ligaduras de unión, matices y cambios de tempo. Las indicaciones dinámicas, crescendi y decrescendi se traducen a niveles de volumen MIDI. Las marcas dinámicas se traducen a una fracción fija del rango de volumen MIDI disponible. Los crescendi y decrescendi hacen que el volumen varía linealmente entre sus dos extremos. El efecto de las indicaciones dinámicas sobre la salida MIDI se puede eliminar por completo, véase Sección 3.5.2 [El bloque MIDI], página 337.

El tempo inicial y los cambios de tempo posteriores se pueden especificar utilizando la instrucción `\tempo` dentro de la propia notación musical. Se reflejan en cambios de tempo en la salida MIDI. Esta instrucción normalmente hace que se imprima la indicación metronómica, pero esto se puede eliminar, véase [Indicaciones metronómicas], página 142. Más abajo se describe una forma alternativa de especificar el tempo MIDI inicial o global, véase Sección 3.5.2 [El bloque MIDI], página 337.

Nombres de instrumentos

El nombre del instrumento MIDI se establece mediante la propiedad `Staff.midiInstrument`. El nombre del instrumento se debe elegir de entre los que están en la lista que aparece en Sección B.4 [Instrumentos MIDI], página 459.

```
\new Staff {
  \set Staff.midiInstrument = #"glockenspiel"
  ...notes...
}
\new Staff \with {midiInstrument = #"cello"} {
  ...notes...
}
```

Si el instrumento elegido no coincide exactamente con uno de los instrumentos de la lista de instrumentos MIDI, se usará el instrumento Piano de Cola (`"acoustic grand"`).

Fragmentos de código seleccionados
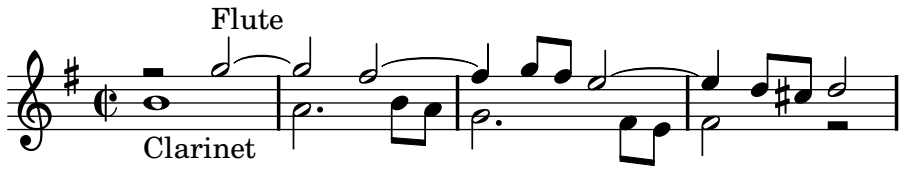
Modificar la salida MIDI para que tenga un canal por cada voz

Al producir una salida MIDI, el comportamiento predeterminado es que cada pentagrama representa un canal MIDI, con todas las voces de dicho pentagrama mezcladas. Esto reduce al mínimo el riesgo de que se agote el número de canales MIDI disponibles, pues existe un máximo de 16 canales por pista.

Sin embargo, cuando se traslada el interpretador `Staff_performer` al contexto `Voice`, cada voz de un pentagrama puede tener su propio canal MIDI, como se muestra en el siguiente ejemplo: a pesar de estar sobre el mismo pentagrama, se crean dos canales MIDI, cada uno con un `midiInstrument` distinto.

```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = #"flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = #"clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
```

```
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 2)
    }
  }
}
```



Advertencias y problemas conocidos

Los cambios del volumen MIDI sólo tienen lugar al principio de la nota, por lo que los crescendi y decrescendi no pueden afectar al volumen de una sola nota.

No todos los repoductores de MIDI manejar adecuadamente los cambios de tempo de la salida MIDI. Entre los reproductores que se sabe que funcionan se encuentran el Reproductor de Medios de MS Windows y timidity.

3.5.2 El bloque MIDI

Si se requiere una salida MIDI, debe aparecer un bloque `\midi` dentro de un bloque de partitura. Es análogo al bloque layout de diseño de página, pero algo maś simple. Con frecuencia, el bloque `\midi` se deja vacío, pero puede contener reorganizaciones de contextos, definiciones de contexto nuevas o código para fijar valores de propiedades. Por ejemplo, en el ejemplo siguiente se establece el tempo inicial exportado alarchivo MIDI sin hacer que se imprima una indicación de tempo:

```
\score {
  ...música...
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 4)
    }
  }
}
```

En este ejemplo, el tempo se establece a 72 pulsos de negra por minuto. Esta clase de especificación de tempo no puede tomar como argumento una figura con puntillo. Si necesitamos una figura con puntillo, dividimos la nota con puntillo en figuras más pequeñas. Por ejemplo, un tempo de 90 negras con puntillo por minuto se puede especificar como 270 corcheas por minuto:

```
tempoWholesPerMinute = #(ly:make-moment 270 8)
```

Las definiciones de contexto siguen con precisión la misma sintaxis que la de un bloque `\layout`. Los módulos de traducción para el sonido reciben el nombre de ejecutantes. Los contextos para la salida MIDI se definen en '`../ly/performer-init.ly`', véase Sección "Other sources of information" in *Manual de Aprendizaje*. Por ejemplo, para eliminar el efecto de los matices de la salida MIDI, inserte las líneas siguientes en el bloque `\midi{ }`.

```
\midi {
  ...
  \context {
    \Voice
    \remove "Dynamic_performer"
```

```
  }
}
```

La salida MIDI se crea sólo cuando se incluye un bloque `\midi` dentro de un bloque de partitura definido con una instrucción `\score`. Si se sitúa dentro de un contexto de partitura instanciado explícitamente (es decir, dentro de un bloque `\new Score`) el archivo no funciona. Para resolverlo, encierre las instrucciones `\new Score` y `\midi` dentro de un bloque `\score`.

```
\score {
  \new Score { ...notas... }
  \midi { }
}
```

3.5.3 ¿Qué sale por el MIDI?

Contemplado en el MIDI

Los siguientes elementos de notación se reflejan en la salida MIDI:

- Alturas
- Cuartos de tono (véase [Alteraciones accidentales], página 4. La reproducción necesita un reproductor que contemple la curvatura de tono o *pitch bend*.)
- Acordes escritos como nombres de acorde
- Ritmos escritos como duraciones de las notas, incluidos los grupos de valoración especial
- Trémolos escritos sin ':[*número*]'
- Ligaduras de unión
- Matices o indicaciones de dinámica
- Crescendi y decrescendi sobre varias notas
- Cambios de tempo insertados con una marca metronómica
- Letra de las canciones

No contemplado en el MIDI

Los siguientes elementos de notación no tienen ningún efecto sobre la salida MIDI:
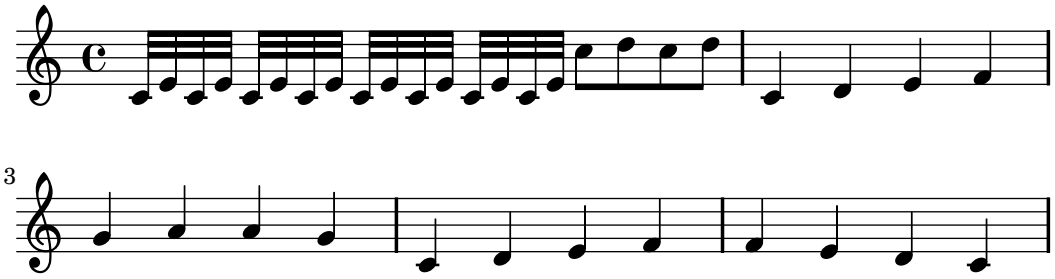
- Duraciones escritas como anotaciones, p.ej. el swing
- Cambios de tempo escritos como anotacioens sin marca de tempo
- Staccato y otras articulaciones y ornamentos
- Ligaduras de expresión y de fraseo
- Crescendi y decrescendi sobre una nota única
- Trémolos introducidos con ':[*número*]'
- Bajo cifrado

3.5.4 Repeticiones y MIDI

Con un poco de trucaje, se puede hacer que cualquier tipo de repetición esté presente en la salida MIDI. Esto se consigue mediante la aplicación de la función musical `\unfoldRepeats`. Esta función cambia todas las repeticiones a repeticiones desplegadas.

```
\unfoldRepeats {
  \repeat tremolo 8 {c'32 e' }
  \repeat percent 2 { c''8 d'' }
  \repeat volta 2 {c'4 d' e' f'}
  \alternative {
    { g' a' a' g' }
```

```
    {f' e' d' c' }
  }
}
\bar "|."
```



Al crear un archivo de partitura que use `\unfoldRepeats` para el MIDI, es necesario hacer dos bloques `\score`: uno para el MIDI (con repeticiones desplegadas) y otro para la notación impresa (con repeticiones de volta –primera y segunda vez–, tremolo –trémolo–, y percent –repetición de compases–). Por ejemplo,

```
\score {
  ..música..
  \layout { .. }
}
\score {
  \unfoldRepeats ..música..
  \midi { .. }
}
```

3.5.5 Control de los matices en el MIDI

La dinámica en el MIDI está implementada por medio del intérprete Dynamic_performer que reside de forma predeterminada dentro del contexto Voice. Es posible controlar el volumen MIDI general, el volumen relativo de los matices dinámicos y el volumen relativo de los distintos instrumentos.

Indicaciones dinámicas

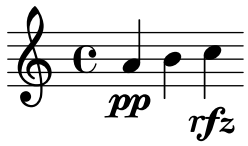
Los matices o indicaciones de dinámica se traducen a una fracción fija del rango de volumen MIDI disponible. Las fracciones predeterminadas se extienden desde 0.25 para *ppppp* hasta 0.95 para *fffff*. El conjunto de indicaciones dinámicas y las fracciones asociadas pueden verse en '`../scm/midi.scm`', véase Sección "Other sources of information" in *Manual de Aprendizaje*. Este conjunto de fracciones se puede cmbiar o extender proporcionando una función que toma como argumento una indicación dinámica y devuelve la fracción requerida, y estableciendo `Score.dynamicAbsoluteVolumeFunction` a esta función.

Por ejemplo, si se requiere una indicación dinámica *rinforzando*, `\rfz`, ésta no tiene ningún efecto predeterminado sobre el voumen MIDI, pues no está incluida en el conjunto predeterminado. De igual manera, si se define una indicación dinámica nueva con `make-dynamic-script`, ésta tampoco estará incluida en el conjunto predeterminado. El ejemplo siguiente muestra cómo hay que añadir el volumen MIDI para estas indicaciones dinámicas. La función Scheme establece la fracción a 0.9 si se encuentra una indicación dinámica rfz, y en caso contrario llama a la función predeterminada.

```
#(define (myDynamics dynamic)
    (if (equal? dynamic "rfz")
      0.9
```

```
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = #"cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
       \relative c'' {
         a\pp b c-\rfz
       }
    }
  }
  \layout {}
  \midi {}
}
```



Como alternativa, si se necesitara redefinir la tabla completa de fracciones, sería mejor usar el procedimiento *default-dynamic-absolute-volume* que aparece en '`../scm/midi.scm`' y la tabla asociada como modelo. El último ejemplo de esta sección muestra cómo se puede hacer esto.

Volumen maestro MIDI
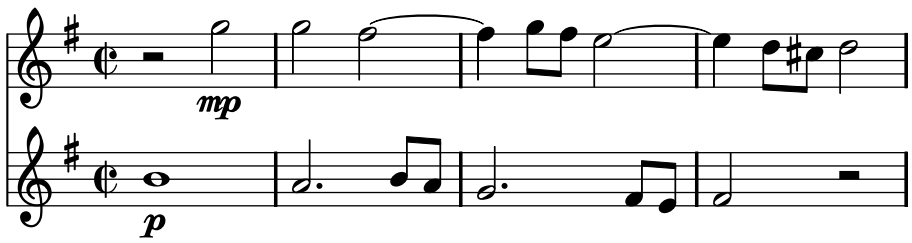
El volumen general máximo y mínimo de las indicaciones dinámicas de MIDI se controla estableciendo valores para las propiedades `midiMinimumVolume` y `midiMaximumVolume` en el nivel de `Score`. Estas propiedades tienen efecto solamente sobre las indicaciones dinámicas, por ello si queremos que tengan validez desde el comienzo de la partitura, se debe colocar allí una indicación de dinámica. La fracción que corresponde a cada indicación dinámica se modifica con esta fórmula:

```
midiMinimumVolume + (midiMaximumVolume - midiMinimumVolume) * fracción
```

En el ejemplo siguiente, el rango de intensidad dinámica del volumen general MIDI está limitado al intervalo desde 0.2 hasta 0.5.

```
\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Staff.midiInstrument = #"flute"
      \new Voice \relative c''' {
        r2 g\mp g fis ~
        fis4 g8 fis e2 ~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \new Voice \relative c'' {
```

```
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 2)
      midiMinimumVolume = #0.2
      midiMaximumVolume = #0.5
    }
  }
}
```
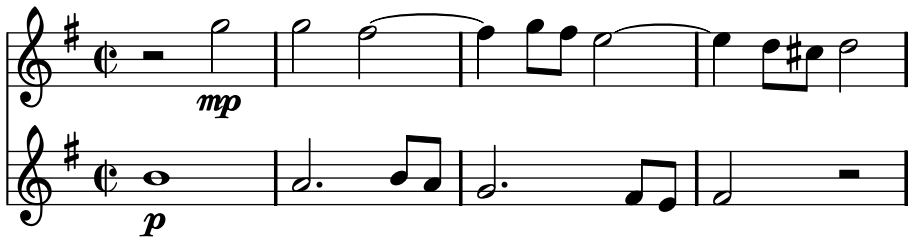


Balance entre instrumentos (i)

Si las propiedades de volumen MIDI máximo y mínimo están establecidas en el contexto `Staff` se pueden controlar los volúmenes relativos de los instrumentos MIDI. Esto proporciona un ecualizador de instrumentos básico que puede realzar significativamente la calidad de la salida MIDI.

En este ejemplo se reduce el volumen del clarinete con relación al de la flauta. Debe haber una indicación dinámica en la primera nota de cada instrumento para que esto funcione correctamente.

```
\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Staff.midiInstrument = #"flute"
      \set Staff.midiMinimumVolume = #0.7
      \set Staff.midiMaximumVolume = #0.9
      \new Voice \relative c''' {
        r2 g\mp g fis ~
        fis4 g8 fis e2 ~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \set Staff.midiMinimumVolume = #0.3
```

```
      \set Staff.midiMaximumVolume = #0.6
      \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 2)
    }
  }
}
```



Balance entre instrumentos (i)

Si las propiedades de volumen MIDI máximo y mínimo no están establecidas, de forma predeterminada LilyPond aplica un pequeño grado de ecualización a algunos instrumentos. Los instrumentos y la ecualización que se aplica se muestra en la tabla *instrument-equalizer-alist* en '`../scm/midi.scm`'.

El ecualizador básico predeterminado se puede sustituir estableciendo `instrumentEqualizer` en el contexto `Score` a un procedimiento Scheme nuevo que acepte el nombre de un instrumento MIDI como único argumento y devuelva un par de fracciones que dan los volúmenes máximo y mínimo que se aplican a dicho instrumento. Esta sustitución se hace de la misma forma que mostramos para reestablecer la fracción de volumen absoluto `dynamicAbsoluteVolumeFunction` al comienzo de esta sección. El ecualizador predeterminado, *default-instrument-equalizer*, en '`../scm/midi.scm`', muestra cómo se puede escribir tal procedimiento.

El ejemplo siguiente establece los volúmenes relativos de la flauta y del clarinete a los mismos valores que el ejemplo anterior.
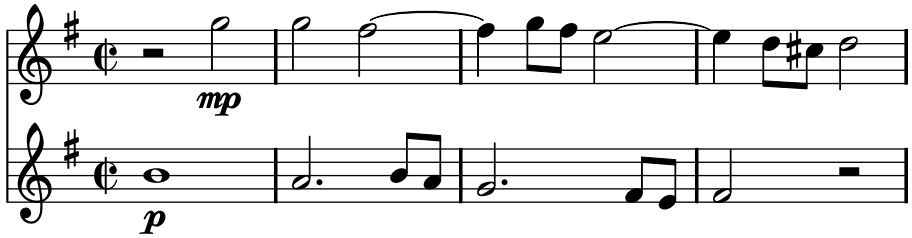
```
#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(
      ("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
```

```
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = #"flute"
      \new Voice \relative c''' {
        r2 g\mp g fis ~
        fis4 g8 fis e2 ~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 2)
    }
  }
}
```



3.5.6 Percusión en MIDI

La notación de los instrumentos de percusión se realiza normalmente en un contexto `DrumStaff` y cuando se hace de esta forma se les da salida correctamente por el canal MIDI 10, pero ciertos instrumentos de percusión de altura determinada, como el xilófono, marimba, vibráfono, timbales, etc., se tratan como instrumentos "normales" y la música para estos instrumentos se debe introducir en contestos de `Staff` normales, no en un contexto `DrumStaff`, para obtener la salida MIDI correcta.

Ciertos sonidos de altura indeterminada que están incluidos en el estándar General MIDI, como el tom melódico, el tambor taiko, los tambores sintéticos, etc., no se pueden acceder a

través del canal MIDI 10, por lo que la notación para dichos instrumentos se debe introducir también en un contexto normal de `Staff`, utilizando las alturas normales adecuadas.

Muchos instrumentos de percusión no están incluidos dentro del estándar General MIDI standard, p.ej. las castañuelas. El método más fácil (aunque no satisfactorio) de producir una salida MIDI al escribir para estos instrumentos es sustituirlos por el sonido más parecido del conjunto estándar.

Advertencias y problemas conocidos

Debido a que el estándar General MIDI no contiene golpes de aro (*rim shots*), para este propósito se utiliza en su lugar el golpe lateral de baqueta (*sidestick*).

4 Problemas de espaciado

La disposición global sobre el papel viene determinada por tres factores: el diseño de página, los saltos de línea y el espaciado. Estos factores influyen entre sí. La elección de uno u otro espaciado determina qué densidad termina teniendo cada sistema de música. Esto influye en el lugar en que se eligen los saltos de línea, y en último término, en el número de páginas que tiene una partitura.

Considerado globalmente, este proceso se produce en cuatro fases: en primer lugar se eligen unas distancias elásticas o 'muelles', basados en la duración de las figuras. Se prueban todas las combinaciones de saltos de línea posibles, y para cada una de ellas se calcula una puntuación de 'maldad'. Después se hace una estimacion de la altura de cada uno de los posibles sistemas. Finalmente se selecciona una combinación de saltos de página y de línea de tal forma que ni el espaciado horizontal ni el vertical estén demasiado estrechos ni estirados.

Los ajustes que influyen en la disposición se pueden agrupar en dos bloques. El bloque `\paper {...}` se sitúa fuera de los bloques `\score {...}` y contiene ajustes relacionados con el documento como un todo. El bloque `\layout {...}` se sitúa dentro de un bloque `\score {...}` y contiene ajustes para dicha partitura en concreto. Si sólo tiene un bloque `\score {...}`, los dos tienen el mismo efecto. En general las instrucciones que se muestran en este capítulo se pueden poner en cualquiera de los dos.

4.1 El papel y las páginas

Esta sección se ocupa de los límites que definen la zona del papel en que se puede imprimir la música.

4.1.1 Tamaño del papel

Existen dos funciones para cambiar el tamaño del papel: `set-default-paper-size` (establecer tamaño de página predeterminado) y `set-paper-size` (establecer tamaño de página). `set-default-paper-size` se debe poner en el ámbito del nivel superior, y `set-paper-size` se debe poner en un bloque `\paper`:

```
#(set-default-paper-size "a4")

\paper {
  #(set-paper-size "a4")
}
```

`set-default-paper-size` establece el tamaño de todas las páginas, mientras que `set-paper-size` establece sólo el tamaño de las páginas a las que se aplica el bloque `\paper`. Por ejemplo, si el bloque `\paper` está al principio del archivo, aplica el tamaño de papel a todas las páginas. Si el bloque `\paper` está dentro de un bloque `\book`, entonces el tamaño del papel sólo se aplica a ese libro.

Se encuentran disponibles los tamaños de papel más comunes, entre ellos `a4`, `letter`, `legal` y `11x17` (conocido también como tabloide). Están contemplados muchos más tamaños de papel de forma predeterminada. Para ver más detalles, consulte '`scm/paper.scm`' y busque la definición de `paper-alist`.

> **Nota:** El tamaño predeterminado del papel es `a4`.

Se pueden añadir tamaños adicionales editando la definición de `paper-alist` en el archivo de inicio '`scm/paper.scm`', aunque se sobreescribirán en la siguiente reinstalación del programa.

Si se añade el símbolo `'landscape` (apaisado) como argumento a `set-default-paper-size`, las páginas se giran 90 grados y se establece en consonancia una mayor longitud de las líneas.

```
#(set-default-paper-size "a6" 'landscape)
```

Al establecer el tamaño del papel se ajustan un cierto número de variables del `\paper`, tales como los márgenes. Para utilizar un tamaño de página determinado con variables de `\paper` alteradas, establezca el tamaño de página antes de dar valores a las variables.

Véase también

Archivos instalados: '`scm/paper.scm`'.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

4.1.2 Formateo de las páginas

Los márgenes, encabezamientos y pies de página y otras variables de diseño de página se establecen automáticamente de acuerdo con el tamaño del papel.

Esta sección relaciona y describe cierto número de variables del papel que se pueden alterar.

Dimensiones verticales

Estas variables se utilizan para establecer las diferentes dimensiones verticales de la página:

`after-title-space`
: Espacio entre el título y el primer sistema. Predeterminado: `5\mm`.

`before-title-space`
: Distancia entre el último sistema de la pieza anterior y el título de la siguiente. Predeterminado: `10\mm`.

`between-system-padding`
: Distancia mínima que se deja siempre entre el símbolo más bajo de un sistema y el más alto del siguiente. Predeterminado: `4\mm`.

  El aumento de este valor separa más los sistemas cuyas cajas circundantes casi se tocan.

`between-system-space`
: Distancia entre los sistemas. Es la distancia ideal desde el centro del pentagrama inferior de un sistema hasta el centro del pentagrama superior del sistema siguiente. Predeterminado: `20\mm`.

  El aumento de este valor da a la página un aspecto más regular a costa de utilizar más espacio vertical.

`between-title-space`
: Distancia entre títulos consecutivos (p.ej., el título del libro y el título de una pieza). Predeterminado: `2\mm`.

`bottom-margin`
: Margen entre el pie de página y la parte inferior del papel. Predeterminado: `6\mm`.

`foot-separation`
: Distancia entre el sistema situado más abajo y el pie de página. Predeterminado: `4\mm`.

`head-separation`
: Distancia entre el sistema situado más arriba y el encabezamiento de la página. Predeterminado: `4\mm`.

`page-top-space`
: Distancia desde la parte alta del área imprimible hasta el centro del primer pentagrama. Esto sólo funciona para pentagramas con una anchura vertical pequeña. Los pentagramas grandes se establecen con la parte alta de su caja circundante alineada a la parte alta del área imprimible. Predeterminado: `12\mm`.

`paper-height`
: Altura de la página. Predeterminado: altura del tamaño actual del papel. Para ver más detalles, consulte Sección 4.1.1 [Tamaño del papel], página 345.

`top-margin`
: Margen entre el encabezamiento y la parte alta del papel. Predeteminado: `5\mm`.

Fragmentos de código seleccionados

El encabezamiento y el pie se crean por parte de las funciones `make-footer` y `make-header`, definidas dentro de `\paper`. Las implementaciones predeterminadas están en '`ly/paper-defaults.ly`' y en '`ly/titling-init.ly`'.

El diseño de la página en sí está realizado por dos funciones dentro del bloque `\paper`, `page-music-height` y `page-make-stencil`. El primero informa al algoritmo de saltos de línea de la cantidad de espacio que está disponible en una página, y el último crea la propia página dado el sistema que poner sobre ella.

Se pueden definir valores del bloque `\paper` en Scheme. En tal caso, `mm`, `in`, `pt`, y `cm` son variables que están definidas en '`paper-defaults.ly`' con valores en milímetros. Por ello el valor 2 cm se debe multiplicar en el siguiente ejemplo:

```
\paper {
 #(define bottom-margin (* 2 cm))
}
```

Ejemplo:

```
\paper{
  paper-width = 2\cm
  top-margin = 3\cm
  bottom-margin = 3\cm
  ragged-last-bottom = ##t
}
```

Este segundo ejemplo centra los números de página en la parte baja del papel.

```
\paper {
  print-page-number = ##t
  print-first-page-number = ##t
  oddHeaderMarkup = \markup \fill-line { " " }
  evenHeaderMarkup = \markup \fill-line { " " }
  oddFooterMarkup = \markup { \fill-line {
     \bold \fontsize #3 \on-the-fly #print-page-number-check-first
     \fromproperty #'page:page-number-string } }
  evenFooterMarkup = \markup { \fill-line {
     \bold \fontsize #3 \on-the-fly #print-page-number-check-first
     \fromproperty #'page:page-number-string } }
}
```

Véase también

Referencia de la notación: Sección 4.4.2 [Espaciado vertical entre sistemas], página 362.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

Dimensiones horizontales

Nota: Si se establece `paper-width` manualmente, `line-width`, `left-margin`, `indent` y `short-indent` pueden necesitar ajustarse también.

Existe un cierto número de variables que determinan las dimensiones horizontales de una página:

`horizontal-shift`
: Medida en que todos los sistemas (incluidos los títulos de cabecera y los separadores de sistemas) se desplazan a la derecha. Predeterminado: `0.0`.

`indent`
: Nivel de sangrado para el primer sistema de una partitura. Predeterminado: `paper-width` dividido por `14`, tal y como viene determinado por `set-default-paper-size` o `set-paper-size`.

`left-margin`
: El margen entre el límite izquierdo del papel y el comienzo de los sistemas. Predeterminado: `10\mm`, tal y como se define por parte de `set-default-paper-size` o de `set-paper-size`.

`line-width`
: Ancho de los sistemas. Predeterminado: `paper-width` menos `20\mm`, como viene determinado por `set-default-paper-size` o por `set-paper-size`.

`paper-width`
: Ancho de la página. Predeterminado: la anchura del tamaño actual del papel. Para ver más detalles, consulte Sección 4.1.1 [Tamaño del papel], página 345.

`short-indent`
: Nivel de sangrado para todos los sistemas de una partitura excepto el primero. Predeterminado: `0`, como se determina en `set-default-paper-size` o en `set-paper-size`.

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Advertencias y problemas conocidos

La opción `right-margin` está definida pero no establece aún el margen derecho. El valor del margen derecho se debe definir ajustando los valores de `left-margin` y de `line-width`.

Otras variables de diseño de página

Estas variables se pueden usar para ajustar el diseño de la página en general.

`auto-first-page-number`
: El algoritmo de división de páginas está afectado por el hecho de que el número de la primera página sea par o impar. Si está establecido al valor verdadero, el algoritmo de división de páginas decide si comenzar con un número par o impar. Esto hace que el número de la primera página se quede como está, o que se aumente en una unidad. Predeterminado: `##f`.

`blank-last-page-force`
: Penalización por terminar la partitura en una página de numeración impar. Predeterminado: `0`.

`blank-page-force`
: Penalización por tener una página en blanco en medio de una partitura. Esto no se usa por parte de `ly:optimal-breaking` porque éste nunca considera la posibilidad de tener páginas en blanco en mitad de una partitura. Predeterminado: `5`.

`first-page-number`
: Valor del número de la primera página. Predeterminado: `#1`.

`page-breaking-between-system-padding`
: Engaña al divisor de páginas para que crea que `between-system-padding` está establecido a algo distinto de lo que está realmente. Por ejemplo, si esta variable se establece a algo mucho mayor que `between-system-padding`, entonces el divisor de páginas colocará menos sistemas en cada página. Predeterminado: no establecido.

`page-count`
: Número de páginas que usar para una partitura. Predeterminado: sin fijar.

`page-limit-inter-system-space`
: Si tiene un valor verdadero, limita el espacio entre los sistemas de una página en la que sobra mucho espacio. Predeterminado: `##f`. Para ver más detalles, consulte Sección 4.4.2 [Espaciado vertical entre sistemas], página 362.

`page-limit-inter-system-space-factor`
: Factor usado por `page-limit-inter-system-space`. Predeterminado: `1.4`. Para ver más detalles, consulte Sección 4.4.2 [Espaciado vertical entre sistemas], página 362.

`page-spacing-weight`
: Importancia relativa del espacio (vertical) de las páginas y el espaciado (horizontal) de las líneas. Los valores altos hacen que el espaciado de la página tenga más importancia. Predeterminado: `#10`.

`print-all-headers`
: Si está establecido a un valor verdadero, imprime todas las cabeceras para cada una de las `\score` en la salida impresa. Normalmente sólo se imprimen las variables de encabezamiento de pieza y opus. Predeterminado: `##f`.

`print-first-page-number`
: Si está establecido a un valor verdadero, se imprime un número de página en la primera página. Predeterminado: `##f`.

`print-page-number`
: Si está establecido a un valor falso, no se imprimen los números de página. Predeterminado: `##t`.

`ragged-bottom`
: Si está establecido a un valor verdadero, los sistemas no ocuparán verticalmente toda la altura de la página. Esto no afecta a la última página. Predeterminado: `##f`.

  Esto se debería establecer al valor verdadero para piezas que tienen sólo dos o tres sistemas por página, por ejemplo partituras orquestales.

`ragged-last`
: Si está establecido a un valor verdadero, el último sistema de la partitura no llenará la anchura de la línea. En su lugar, el último sistema termina en su longitud horizontal natural. Predeterminado: `##f`.

`ragged-last-bottom`
: Si está establecido a un valor falso, los sistemas se repartirán verticalmente por toda la última página. Predeterminado: `##t`.

  Las piezas que llenan generosamente dos o más páginas deberían tener esto establecido al valor verdadero.

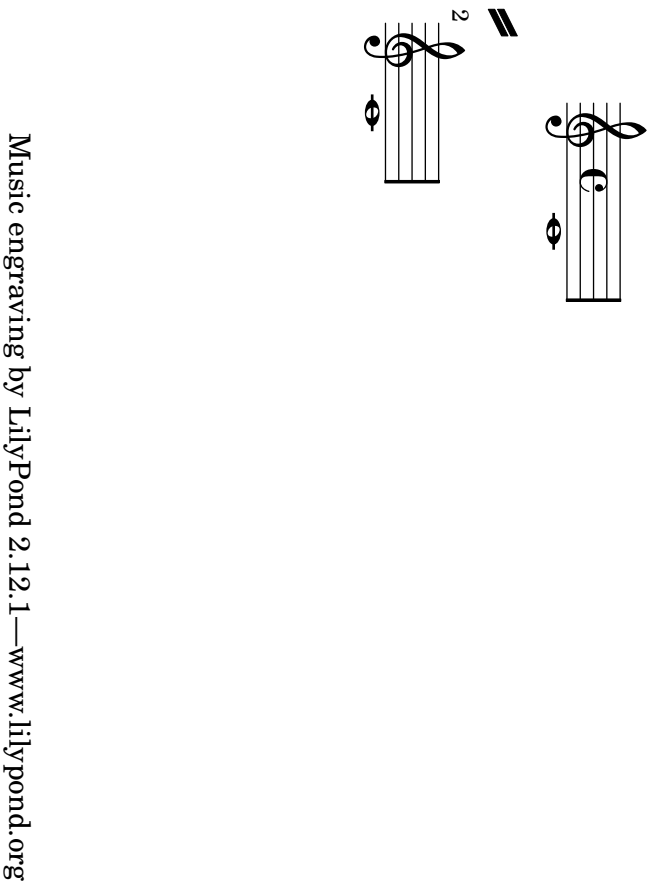
  También afecta a la última página de las partes de libro, es decir, partes de un libro que se han creado con bloques `\bookpart`.

`ragged-right`
: Si está establecido a un valor verdadero, los sistemas no llenarán el ancho de línea disponible. En su lugar, los sistemas terminan en su longitud horizontal natural. Predeterminado: `##f`.

  Si la partitura sólo tiene un sistema, el valor predeterminado es `##t`.

`system-separator-markup`
: Objeto de marcado que se inserta entre los sistemas. Se suele usar para partituras orquestales. Predeterminado: sin establecer.

  Se ofrece la instrucción de marcado `\slashSeparator` como un valor predeterminado apropiado, por ejemplo



`system-count`
: Cantidad de sistemas que usar por parte de la partitura. Predeterminado: sin establecer.

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Advertencias y problemas conocidos

El encabezamiento de página predeterminado pone el número de página y el campo `instrument` del bloque `\header` sobre la misma línea.

Los títulos de cabecera (tomados de la sección `\header{}`) se tratan como un sistema, por lo que `ragged-bottom` y `ragged-last-bottom` añaden espacio entre los títulos y el primer sistema de la partitura.

4.2 Disposición de la música

4.2.1 Establecer el tamaño del pentagrama

El **tamaño de pentagrama** predeterminado se establece en 20 puntos. Esto se puede modificar de dos maneras:

Para establecer globalmente el tamaño del pentagrama para todas las partituras de un archivo (o en un bloque `book`, para ser exactos), utilice `set-global-staff-size`.

```
#(set-global-staff-size 14)
```

Esto establece el tamaño global predeterminado a una altura de pentagrama de 14pt y escala todas las tipografías según corresponda.

Para establecer el tamaño del pentagrama de forma individual para cada partitura, use

```
\score{
  ...
  \layout{
  #(layout-set-staff-size 15)
  }
}
```

La fuente tipográfica Feta ofrece símbolos musicales a ocho tamaños distintos. Cada fuente tipográfica está afinada para un tamaño de pentagrama distinto: a un tamaño menor la tipografía se vuelve más pesada, para que coincida con las líneas de pentagrama relativamente más gruesas. Los tamaños de tipografía recomendados están relacionados en la tabla siguiente:

nombre de la fuente	**altura del pentagrama (pt)**	**altura del pentagrama (mm)**	**usos**
feta11	11.22	3.9	partituras de bolsillo
feta13	12.60	4.4	
feta14	14.14	5.0	
feta16	15.87	5.6	
feta18	17.82	6.3	cancioneros
feta20	20	7.0	particellas estándar
feta23	22.45	7.9	
feta26	25.2	8.9	

Estas tipografías están disponibles en cualquier tamaño. La propiedad de contexto `fontSize` y la propiedad de disposición `staff-space` (en Sección “StaffSymbol” in *Referencia de Funcionamiento Interno*) se pueden usar para afinar el tamaño de los pentagramas individuales. Los tamaños de pentagrama individuales están en relación al tamaño global.

Véase también

Referencia de la notación: [Seleccionar el tamaño de la tipografía para la notación], página 154.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Advertencias y problemas conocidos

`layout-set-staff-size` no cambia la distancia entre las líneas del pentagrama.

4.2.2 Disposición de la partitura

Mientras que `\paper` contiene ajustes relativos al formato de página del documento completo, `\layout` contiene ajustes para la disposición específica de cada partitura.

```
\layout {
  indent = 2.0\cm
  \context { \Staff
    \override VerticalAxisGroup #'minimum-Y-extent = #'(-6 . 6)
  }
  \context { \Voice
    \override TextScript #'padding = #1.0
    \override Glissando #'thickness = #3
  }
}
```

Véase también

Referencia de la notación: Sección 5.1.4 [Cambiar los valores por omisión de los contextos], página 393.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.3 Saltos

4.3.1 Saltos de línea

Normalmente los saltos de línea se determinan automáticamente. Se eligen de forma que las líneas no aparezcan demasiado apretadas ni demasiado sueltas, y que las líneas consecutivas tengan una densidad similar. Ocasionalmente podemos querer sobreescribir los saltos automáticos; podemos hacerlo especficando `\break`. Esto fuerza un salto de línea en ese punto. Sin embargo, los saltos de línea sólo pueden suceder al final de los compases ‘completos’, es decir, donde no queda ninguna nota o grupo especial ‘colgando’ por encima de la línea divisoria. Si queremos poner un salto de línea donde no hay línea divisoria, podemos forzar una barra de compás invisible introduciendo `\bar ""`, aunque de nuevo no deben quedar notas colgando en ninguno de los pentagramas en este punto, o se ignorarán.

La instrucción opuesta, `\noBreak`, prohíbe un salto de línea en la barra divisoria en que se inserta.

Los ajustes más básicos que influyen sobre el espaciado de las líneas son `indent` y `line-width`. Se establecen dentro del bloque `\layout`. Controlan el sangrado de la primera línea de música, y la longitud de las líneas.

Si se establece `ragged-right` a verdadero en el bloque `\layout`, los sistemas terminan en su longitud horizontal natural, en lugar de distribuirse horizontalmente para llenar toda la línea. Esto es de utilidad para fragmentos cortos, y para efectuar una comprobación de lo apretado que es el espaciado natural.

La opción `ragged-last` es similar a `ragged-right`, pero afecta sólo a la última línea de la pieza.

```
\layout {
indent = #0
line-width = #150
ragged-last = ##t
}
```

Para saltos de línea a intervalos regulares utilice `\break` separado mediante desplazamientos con `\skip` y repetidos con `\repeat`. Por ejemplo, esto haría que los 28 compases siguientes (suponiendo un compás de 4/4) se dividan cada cuatro compases, y sólo en dichos lugares:

```
<< \repeat unfold 7 {
         s1 \noBreak s1 \noBreak
         s1 \noBreak s1 \break }
   the real music
>>
```

Una configuración de división de líneas se puede guardar como archivo `.ly` automáticamente. Ello permite que alineaciones verticales se estiren para que encajen en las páginas durante una segunda ejecución del proceso de formateo. Esta posibilidad es bastante novedosa y difícil. Hay más detalles disponibles en Sección “Spacing” in *Fragmentos de código*.

Instrucciones predefinidas

`\break`, `\noBreak`.

Véase también
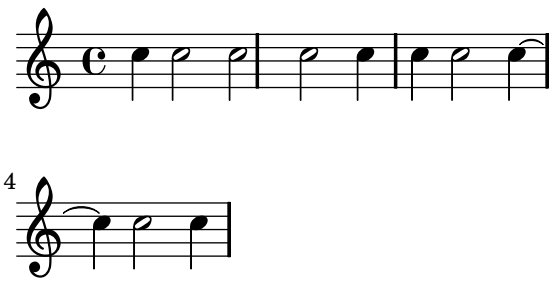
Referencia de funcionamiento interno: Sección “LineBreakEvent” in *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Advertencias y problemas conocidos

Sólo pueden ocurrir saltos de línea si hay una ‘buena’ línea divisoria. Una nota colgando de una línea divisoria no es adecuada, como

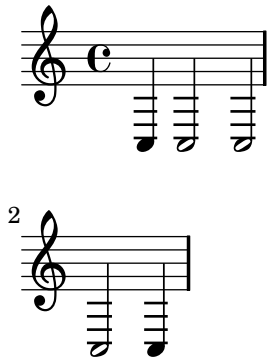
```
c4 c2 << c2 {s4 \break } >>  % esto no hace nada
c2 c4 |           % un salto aquí funcionaría
c4 c2 c4 ~ \break % como por ejemplo este salto
c4 c2 c4
```



Esto se puede evitar eliminando el grabador `Forbid_line_break_engraver`. Observe que los saltos de línea forzados manualmente se tienen que añadir en paralelo con la música.

```
\new Voice \with {
  \remove Forbid_line_break_engraver
} {
  c4 c2 << c2 {s4 \break } >>  % ahora se permite el salto de línea
  c2 c4
```

```
}
```



De forma parecida, los saltos de línea están prohibidos normalmente cuando las barras cruzan a las líneas divisorias. Este comportamiento se puede modificar con el establecimiento de `\override Beam #'breakable = ##t`.

4.3.2 Saltos de página

Se puede sobreescribir el mecanismo predeterminado de salto de página insertando instrucciones `\pageBreak` o `\noPageBreak`. Estas instrucciones son análogas a `\break` y `\noBreak`. Se deben insertar en una línea divisoria. Estas instrucciones fuerzan y prohíben, respectivamente, la eventualidad de un salto de página. Por supuesto, la instrucción `\pageBreak` también fuerza un salto de línea.

Las instrucciones `\pageBreak` y `\noPageBreak` también se pueden insertar en el nivel más alto, entre las partituras y los elementos de marcado situados en el nivel superior.

Hay ajustes análogos a `ragged-right` y `ragged-last` que tienen el mismo efecto sobre el espaciado vertical: `ragged-bottom` y `ragged-last-bottom`. Si están establecidos a `##t` los sistemas de todas las páginas o sólo de la última página, respectivamente, no se verán justificados verticalmente.

Para ver más detalles, consulte Sección 4.4 [Espaciado vertical], página 360.

Los saltos de página se calculan por medio de la función `page-breaking`. LilyPond ofrece tres algoritmos para el cómputo de los saltos de página: `ly:optimal-breaking`, `ly:page-turn-breaking` y `ly:minimal-breaking`. El predeterminado es `ly:optimal-breaking`, pero el valor se puede cambiar en el bloque `\paper`:

```
\paper{
  #(define page-breaking ly:page-turn-breaking)
}
```

El antiguo algoritmo de división de páginas se llama `optimal-page-breaks`. Si tiene problemas con los nuevos divisores de página, puede habilitar el antiguo como forma de rodear el problema.

Cuando un libro tiene muchas partituras y páginas, puede ser difícil resolver el problema de los saltos de página, necesitando mucha memoria y prolongados tiempos de procesamiento. Para facilitar el proceso de división en páginas, se usan los bloques `\bookpart` para dividir el libro en varias partes: los saltos de página se producen de manera independiente en cada parte. También se pueden usar diferentes funciones de división en páginas para las distintas partes del libro.

```
\bookpart {
  \header {
    subtitle = "Prefacio"
  }
  \paper {
```

```
    %% En una parte que consiste en texto principalmente,
    %% puede ser preferible ly:minimal-breaking
    #(define page-breaking ly:minimal-breaking)
  }
  \markup { ... }
  ...
}
\bookpart {
  %% En esta parte, consistente en música, se usa la función
  %% óptima predeterminada de saltos de página.
  \header {
    subtitle = "Primer movimiento"
  }
  \score { ... }
  ...
}
```

Instrucciones predefinidas

`\pageBreak`, `\noPageBreak`.

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.3.3 Saltos de página óptimos

La función `ly:optimal-breaking` es el método predeterminado de LilyPond para determinar los saltos de página. Intenta hallar una división de páginas que haga mínimos el apretujamiento y la distensión, tanto horizontal como verticalmente. A diferencia de `ly:page-turn-breaking`, no tiene un concepto de los pasos de página.

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.3.4 Paso de página óptimo

Con frecuencia es necesario encontrar una configuración de división de páginas de manera que haya un silencio al final de una página de cada dos. De esta forma, el músico puede pasar la página sin perder notas. La función `ly:page-turn-breaking` trata de encontrar una división de páginas que haga mínimos el apretujamiento y el estiramiento, pero con la restricción añadida de que sólo se permite introducir vueltas de página en los lugares especificados.

Hay dos etapas en el uso de esta función de división de páginas. En primer lugar debemos habilitarlo en el bloque `\paper`, como se explicó en Sección 4.3.2 [Saltos de página], página 354. Entonces debemos decirle a la función dónde nos gustaría permitir los saltos de página.

Hay dos formas de conseguir la segunda tarea. Primero, podemos especificar manualmente cada uno de los pasos de página potenciales, insertando `\allowPageTurn` en nuestro archivo de entrada en los lugares adecuados.

Si esto es demasiado tedioso, podemos añadir un grabador `Page_turn_engraver` a un contexto Staff o Voice. El grabador `Page_turn_engraver` analizará el contexto en biusca de secciones sin notas (observe que no busca silencios, sino la ausencia de notas. Se hace así para que la polifonía en un solo pentagrama con silencios en una de las voces no arruine la labor del grabador `Page_turn_engraver`). Cuando encuentra una sección sin notas suficientemente larga, el grabador `Page_turn_engraver` inserta un `\allowPageTurn` en la última barra de compás de

dicha sección, a no ser que haya una barra 'especial' de compás (como una doble barra), en cuyo caso se insertará el `\allowPageTurn` en la última barra 'especial' de compás de la sección.

El grabador `Page_turn_engraver` lee la propiedad de contexto `minimumPageTurnLength` para determinar qué longitud debe tener una sección sin notas antesd e que se considere la posibilidad de un paso de página. El valor predeterminado para `minimumPageTurnLength` es `#(ly:make-moment 1 1)`. Si quiere inhabilitar las vueltas de página, puede establecerlo a algún valor muy grande.

```
\new Staff \with { \consists "Page_turn_engraver" }
{
  a4 b c d |
  R1 | % a page turn will be allowed here
  a4 b c d |
  \set Staff.minimumPageTurnLength = #(ly:make-moment 5 2)
  R1 | % a page turn will not be allowed here
  a4 b r2 |
  R1*2 | % a page turn will be allowed here
  a1
}
```

El grabador `Page_turn_engraver` detecta las repeticiones de primera y segunda vez. Sólo permite un pase de página durante la repetición si hay suficiente tiempo al principio y al final de la repetición para volver a pasar la página hacia atrás. El grabador `Page_turn_engraver` también puede inhabilitar los pasos de página si la repetición es muy corta. Si establecemos la propiedad de contexto `minimumRepeatLengthForPageTurn` entonces el grabador `Page_turn_engraver` sólo permitirá los pases de página en las repeticiones cuya duración sea mayor que este valor.

Las instrucciones de paso de página, `\pageTurn`, `\noPageTurn` y `\allowPageTurn`, se pueden usar también en el nivel más elevado del código, entre las partituras y los elementos de marcado del nivel superior.

Instrucciones predefinidas

`\pageTurn`, `\noPageTurn`, `\allowPageTurn`.

Véase también

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

Advertencias y problemas conocidos

Sólo debería haber un grabador `Page_turn_engraver` dentro de una partitura. Si hay más de uno, se interferirán entre sí.

4.3.5 Saltos de página mínimos

La función `ly:minimal-breaking` efectúa unos cálculos mínimos para determinar los saltos de página: completa una página con tantos sistemas como sea posible antes de continuar con la siguiente. Así, puede preferirse para partituras con muchas páginas, donde las otras funciones de salto de página pueden resultar demasiado lentas o ávidas de memoria, o con una gran cantidad de textos. Se habilita utilizando:

```
\paper {
  #(define page-breaking ly:minimal-breaking)
}
```

Véase también

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

4.3.6 Saltos de línea explícitos
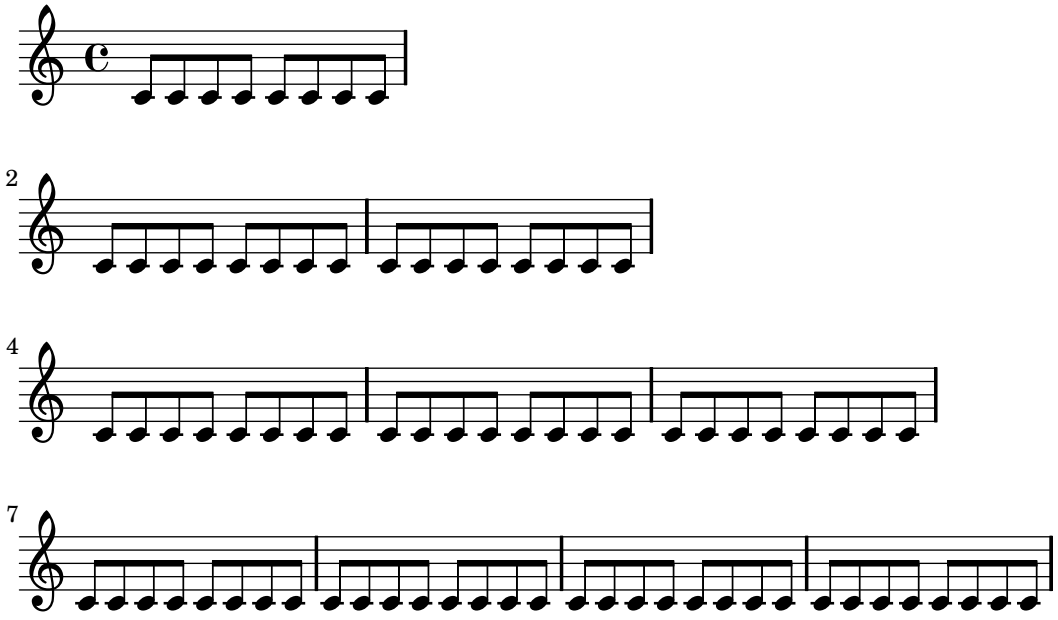
Lily a veces rechaza las instrucciones `\break` y `\pageBreak` explícitas. Hay dos instrucciones para sobreescribir este comportamiento:
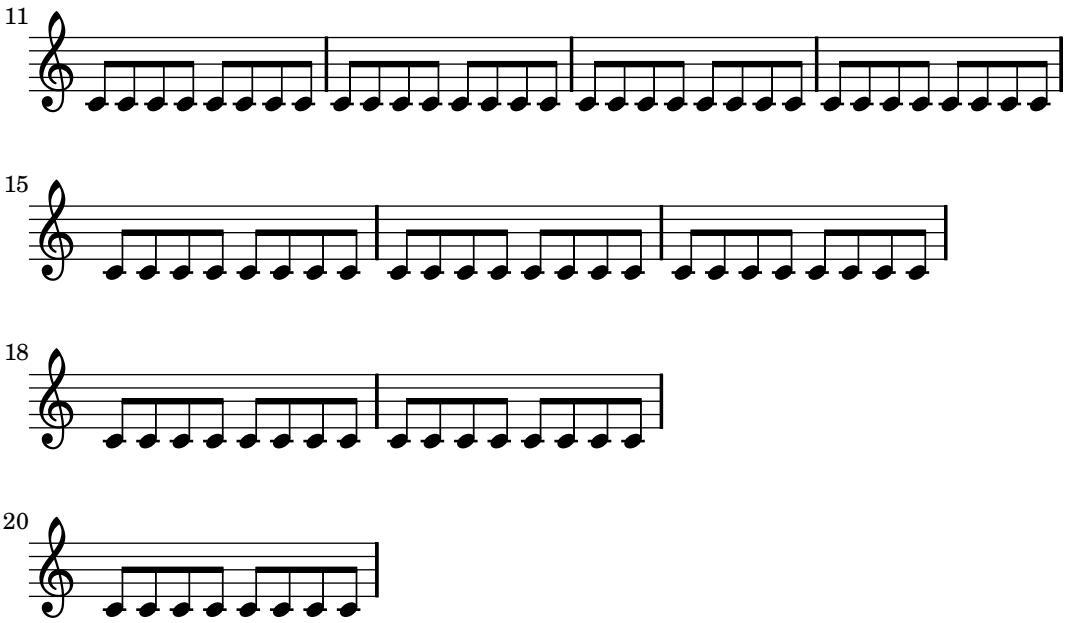
```
\override NonMusicalPaperColumn #'line-break-permission = ##f
\override NonMusicalPaperColumn #'page-break-permission = ##f
```

Cuando se sobreescribe el valor de `line-break-permission` a falso, Lily inserta saltos de línea en las instrucciones `\break` explícitas y en ningún otro lugar. Cuando se sobreescribe el valor de `page-break-permission` a falso, Lily inserta saltos de página en las instrucciones `\pageBreak` explícitas y en ningún otro lugar.

```
\paper {
  indent = #0
  ragged-right = ##t
  ragged-bottom = ##t
}

\score {
  \new Score \with {
    \override NonMusicalPaperColumn #'line-break-permission = ##f
    \override NonMusicalPaperColumn #'page-break-permission = ##f
  } {
    \new Staff {
      \repeat unfold 2 { c'8 c'8 c'8 c'8 } \break
      \repeat unfold 4 { c'8 c'8 c'8 c'8 } \break
      \repeat unfold 6 { c'8 c'8 c'8 c'8 } \break
      \repeat unfold 8 { c'8 c'8 c'8 c'8 } \pageBreak
      \repeat unfold 8 { c'8 c'8 c'8 c'8 } \break
      \repeat unfold 6 { c'8 c'8 c'8 c'8 } \break
      \repeat unfold 4 { c'8 c'8 c'8 c'8 } \break
      \repeat unfold 2 { c'8 c'8 c'8 c'8 }
    }
  }
}
```





Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.3.7 Utilizar una voz adicional para los saltos de línea
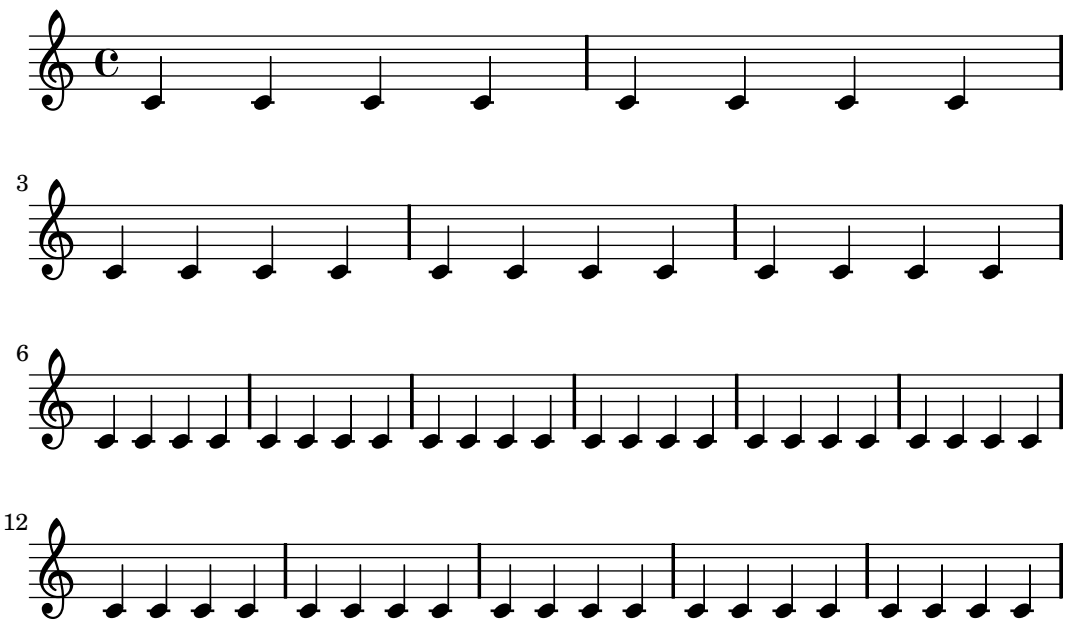
La información sobre saltos de línea y de página suele aparecer directamente intermezclado dentro del código de notas.

```
\new Score {
  \new Staff {
    \repeat unfold 2 { c'4 c'4 c'4 c'4 }
    \break
    \repeat unfold 3 { c'4 c'4 c'4 c'4 }
  }
}
```

Esto hace sencillas de introducir a las instrucciones `\break` y `\pageBreak`, pero mezcla la escritura de música con información que especifica cómo se debe disponer la música sobre la página. Podemos mantener la introducción de música y la información de saltos de línea y de página en dos lugares separados mediante la introducción de una voz adicional que contenga los saltos. Esta voz adicional contiene solamente desplazamientos o ‘skips’ junto con los `\break`, `pageBreak` y otras informaciones sobre la disposición de los saltos.

```
\new Score {
  \new Staff <<
     \new Voice {
        s1 * 2 \break
        s1 * 3 \break
        s1 * 6 \break
        s1 * 5 \break
     }
     \new Voice {
        \repeat unfold 2 { c'4 c'4 c'4 c'4 }
        \repeat unfold 3 { c'4 c'4 c'4 c'4 }
        \repeat unfold 6 { c'4 c'4 c'4 c'4 }
        \repeat unfold 5 { c'4 c'4 c'4 c'4 }
     }
  >>
```

}



Este patrón resulta especialmente útil cuando se sobreescribe `line-break-system-details` y las otras útiles (pero largas) propiedades de `NonMusicalPaperColumnGrob`, como se explica en Sección 4.4 [Espaciado vertical], página 360.
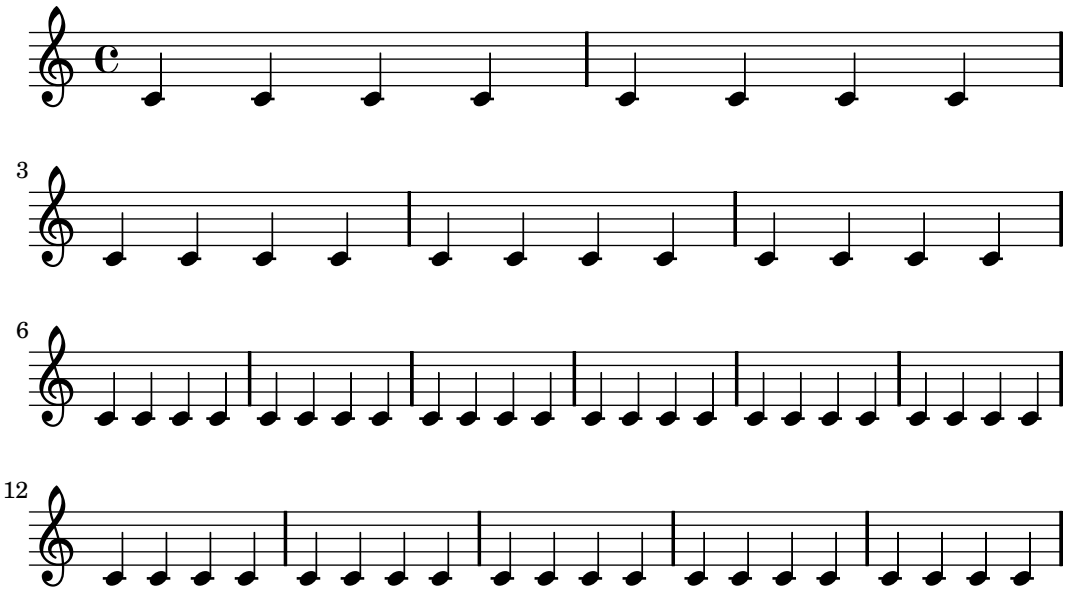
```
\new Score {
  \new Staff <<
     \new Voice {

        \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 0))
        s1 * 2 \break

        \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 35))
        s1 * 3 \break

        \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 70))
        s1 * 6 \break

        \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 105))
        s1 * 5 \break
     }
     \new Voice {
        \repeat unfold 2 { c'4 c'4 c'4 c'4 }
        \repeat unfold 3 { c'4 c'4 c'4 c'4 }
        \repeat unfold 6 { c'4 c'4 c'4 c'4 }
        \repeat unfold 5 { c'4 c'4 c'4 c'4 }
     }
  >>
}
```



Véase también

Referencia de la notación: Sección 4.4 [Espaciado vertical], página 360.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.4 Espaciado vertical

El espaciado vertical está controlado por tres factores: la cantidad de espacio disponible (es decir, el tamaño del papel y los márgenes), la separación entre los sistemas, y la separación entre los pentagramas dentro de un sistema.

4.4.1 Espaciado vertical dentro de un sistema

La altura de cada sistema se determina automáticamente. Para evitar que los pentagramas se superpongan unos encima de otros, se establecen algunas distancias mínimas. Al modificarlas, podemos poner los pentagramas más cerca unos de otros. Esto reduce el espacio que requiere cada sistema y puede resultar en la obtención de más sistemas por página.

Normalmente los pentagramas se apilan verticalmente. Para hacer que los pentagramas guarden una distancia, se rellena su tamaño vertical. Esto se hace con la propiedad `minimum-Y-extent`. Si se aplica a Sección “VerticalAxisGroup” in *Referencia de Funcionamiento Interno*, controla el tamaño de una línea horizontal, como un pentagrama o una línea de letra. `minimum-Y-extent` acepta una pareja de números, de forma que si queremos hacerlo más pequeño que su valor predeterminado `#'(-4 . 4)` entonces podemos establecer

```
\override Staff.VerticalAxisGroup #'minimum-Y-extent = #'(-3 . 3)
```

Esto fija el tamaño vertical del pentagrama en curso a 3 espacios de pentagrama a cada lado de la tercera línea. El valor `(-3 . 3)` se interpreta como un intervalo, donde la línea central es el 0, por lo que el primer número es negativo en general. Los números no tienen que coincidir; por ejemplo, el pentagrama se puede hacer mayor por debajo dándole el valor `(-6 . 4)`.

Después de que se han determinado los saltos de página, se reevalúa el espaciado vertical dentro de cada sistema para llenar la página de forma más regular; si una página tiene más espacio de sobra, se amplían los sistemas para poder llenar este espacio. La magnitud de esta ampliación se puede configurar a través de la propiedad `max-stretch` del grob Sección “VerticalAlignment” in *Referencia de Funcionamiento Interno*. De forma predeterminada, `max-stretch` se establece a cero, inhabilitando la ampliación. Para habilitar la ampliación, un valor prudente para `max-stretch` es `ly:align-interface::calc-max-stretch`.

En ciertas situaciones, podemos querer ampliar la mayor parte de un sistema y al mismo tiempo dejar algunas partes fijas. Por ejemplo, si una parte de piano aparece en el medio de una

partitura orquestal, podemos querer dejar los pentagramas de piano cercanos entre sí mientras se estira el resto de la partitura. Se puede usar la propiedad `keep-fixed-while-stretching` de Sección "VerticalAxisGroup" in *Referencia de Funcionamiento Interno* para conseguirlo. Cuando se establece al valor `##t`, esta propiedad evita que su pentagrama (o línea de letra) se desplace en relación al que está directamente encima de él. En el ejemplo anterior, podríamos sobreescribir `keep-fixed-while-stretching` al valor `##t` en el segundo pentagrama del piano:
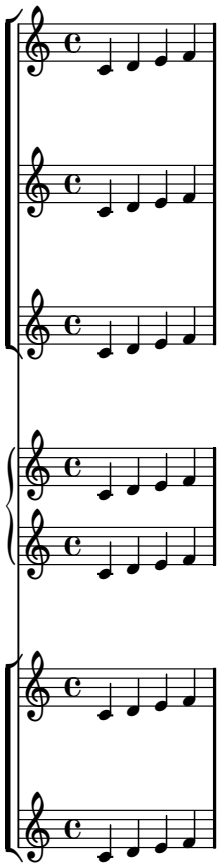
```
#(set-default-paper-size "a6")
#(set-global-staff-size 14.0)

\book {
\paper {
  ragged-last-bottom = ##f
}

\new Score \with
{
  \override VerticalAlignment #'max-stretch = #ly:align-interface::calc-max-stretch
}
{
\new GrandStaff
<<
  \new StaffGroup
  <<
    \new Staff {c' d' e' f'}
    \new Staff {c' d' e' f'}
    \new Staff {c' d' e' f'}
  >>

  \new PianoStaff
  <<
    \new Staff {c' d' e' f'}
    \new Staff \with {
      \override VerticalAxisGroup #'keep-fixed-while-stretching = ##t
    }
    {c' d' e' f'}
  >>

  \new StaffGroup
  <<
    \new Staff {c' d' e' f'}
    \new Staff {c' d' e' f'}
  >>
>>
}
}
```



Music engraving by LilyPond 2.12.1—www.lilypond.org

La alineación vertical de los pentagramas está manejada por el objeto `VerticalAlignment`. Los parámetros de contexto que especifican las dimensiones verticales están descritas en conexión con el grabador `Axis_group_engraver`.

Véase también

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "VerticalAlignment" in *Referencia de Funcionamiento Interno*, Sección "Axis_group_engraver" in *Referencia de Funcionamiento Interno*.

4.4.2 Espaciado vertical entre sistemas

La separación entre los sistemas está controlada por cuatro variables del papel:

```
\paper {
  between-system-space = 1.5\cm
  between-system-padding = #1
  ragged-bottom=##f
  ragged-last-bottom=##f
}
```

Cuando sólo se ponen dos simples sistemas en una página, el espaciado vertical resultante puede ser bastante poco elegante: un sistema en la parte alta de la página y el otro en la parte baja, con un enorme hueco entre ellos. Para evitar esta situación se puede limitar el espacio que se añade entre los sistemas. Esta posibilidad se activa mediante el establecimiento al valor `#t` de la variable `page-limit-inter-system-space` en el bloque `\paper`. La variable del papel `page-limit-inter-system-space-factor` determina en qué cantidad se puede incrementar el
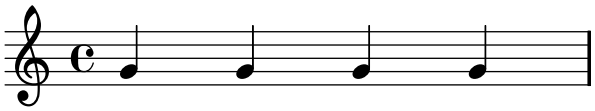
espacio: por ejemplo, el valor `1.3` significa que el espacio puede ser un 30% mayor que lo que sería en una página no justificada hasta abajo.

En el ejemplo siguiente, si el espacio entre los sistemas no estuviese limitado, el segundo sistema de la página 1 se situaría en la parte de abajo de la página. Activando la limitación de espacio, el segundo sistema se coloca más cerca del primero. Estableciendo `page-limit-inter-system-space-factor` a `1`, el espacio sería el mismo que en una página sin justificar por abajo, como la última página.

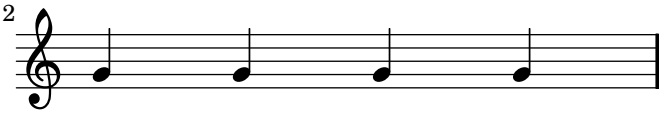
```
#(set-default-paper-size "a6")
\book {
  \paper {
    page-limit-inter-system-space = ##t
    page-limit-inter-system-space-factor = 1.3

    oddFooterMarkup = \markup "page bottom"
    evenFooterMarkup = \markup "page bottom"
    oddHeaderMarkup = \markup \fill-line {
      "page top" \fromproperty #'page:page-number-string }
    evenHeaderMarkup = \markup \fill-line {
      "page top" \fromproperty #'page:page-number-string }
  }
  \new Staff << \repeat unfold 4 { g'4 g' g' g' \break }
                { s1*2 \pageBreak } >>
}
```
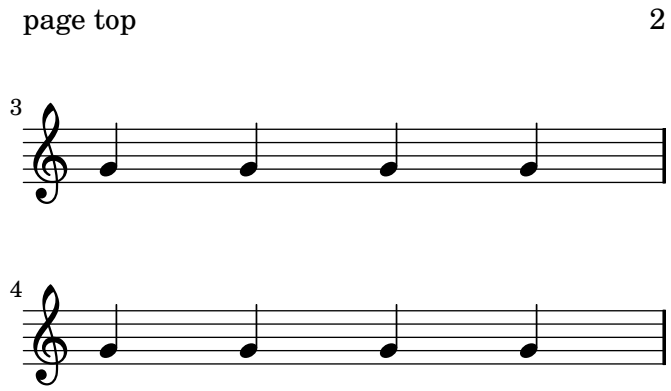
page top 1



2



page bottom



Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.4.3 Posicionamiento explícito de los pentagramas y los sistemas

Una forma de entender los ajustes de `VerticalAxisGroup` y de `\paper` que se explicaron en las dos secciones anteriores es como una colección de distintos ajustes que conciernen principalmente a la cantidad de relleno vertical que adquieren los distintos pentagramas y sistemas al ocupar la página.

Es posible abordar el problema del espaciado vertical de una forma distinta utilizando `NonMusicalPaperColumn #'line-break-system-details`. Donde los ajustes de `VerticalAxisGroup` y de `\paper` especifican el relleno vertical, `NonMusicalPaperColumn #'line-break-system-details` especifica las posiciones verticales exactas en la página.

`NonMusicalPaperColumn #'line-break-system-details` acepta una lista asociativa de cinco ajustes distintos:

- `X-offset`
- `Y-offset`
- `alignment-offsets`
- `alignment-extra-space`
- `fixed-alignment-extra-space`

Las sobreescrituras de los objetos gráficos, entre ellas las de `NonMusicalPaperColumn` que aparece más abajo, pueden ocurrir en tres lugares distintos dentro de un archivo de entrada:

- directamente en medio de las notas
- en un bloque `\context`
- en el bloque `\with`

Cuando sobreescribimos `NonMusicalPaperColumn`, usamos la instrucción `\override` usual en los bloques `\context` y en el bloque `\with`. Por otra parte, cuando sobreescribimos `NonMusicalPaperColumn` en medio de las notas, debemos usar la instrucción especial `\overrideProperty`. He aquí algunas sebreescrituras de `NonMusicalPaperColumn` de ejemplo con la instrucción especial `\overrideProperty`:

```
\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20))
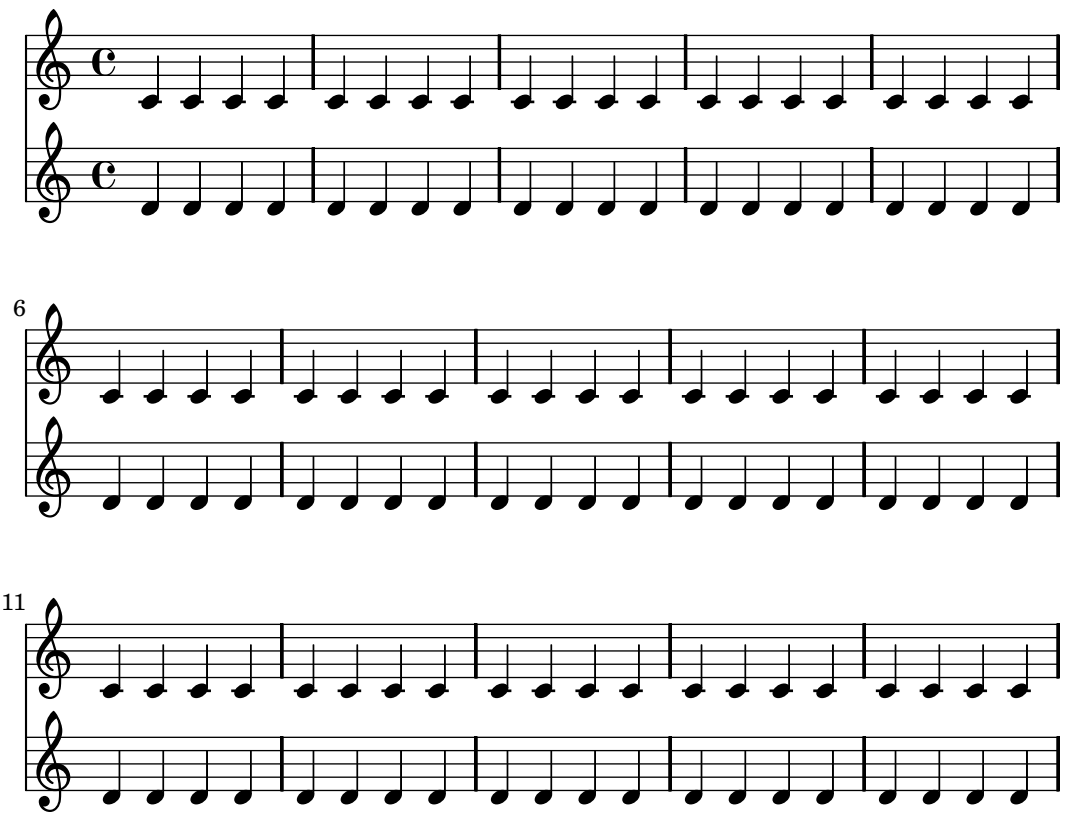
\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((Y-offset . 40))

\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20) (Y-offset . 40))

\override NonMusicalPaperColumn
  #'line-break-system-details #'((alignment-offsets . (0 -15)))
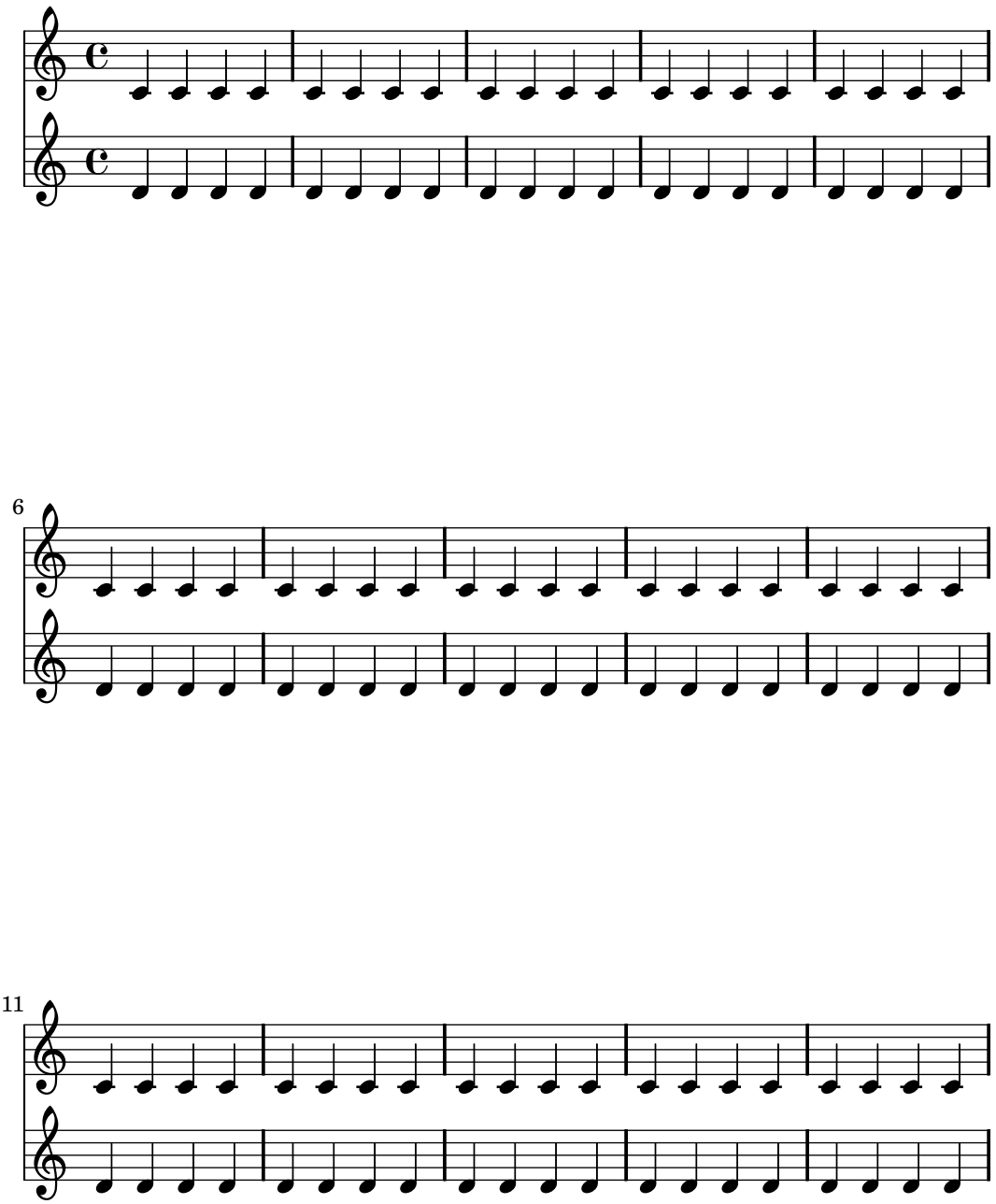
\override NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20) (Y-offset . 40)
                                 (alignment-offsets . (0 -15)))
```

Para entender cómo funciona cada uno de los distintos ajustes, empezamos observando un ejemplo que no incluye absolutamente ninguna sobreescritura.
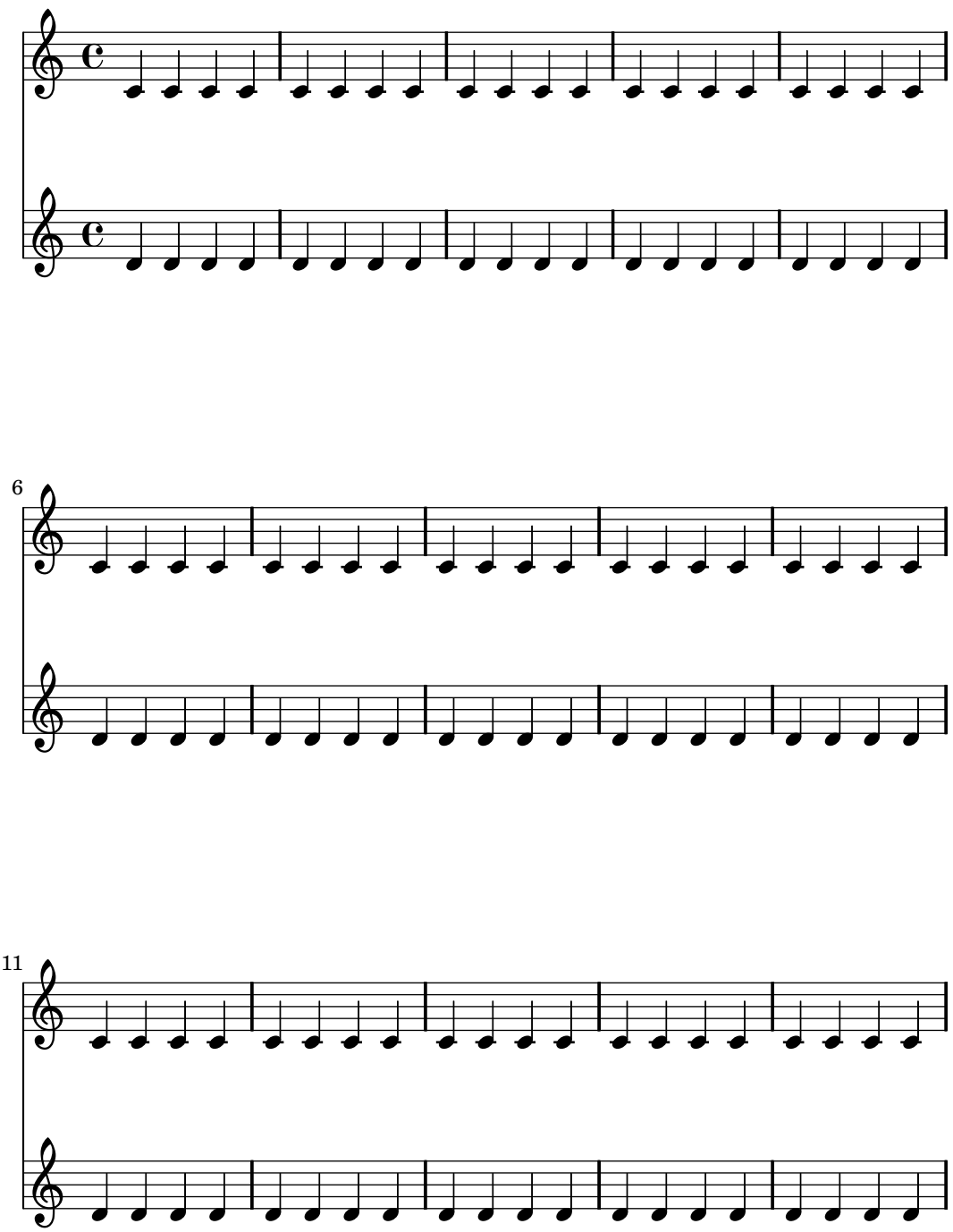


Esta partitura mantiene aislada la información de los saltos de línea y de página en una voz dedicada a ello. Esta técnica de crear una voz de saltos ayuda a mantener separada de la escritura de notas conforme el ejemplo se vuelve más complicado. Consulte Sección 4.3.7 [Utilizar una voz adicional para los saltos de línea], página 358.

Los saltos explícitos dividen la música de forma regular en seis compases por línea. El espaciado vertical es el resultado de los ajustes predeterminados de LilyPond. Para establecer explícitamente el punto de origen vertical de cada sistema, podemos establecer el par `Y-offset` en el atributo `line-break-system-details` del grob (objeto gráfico) `NonMusicalPaperColumn`:

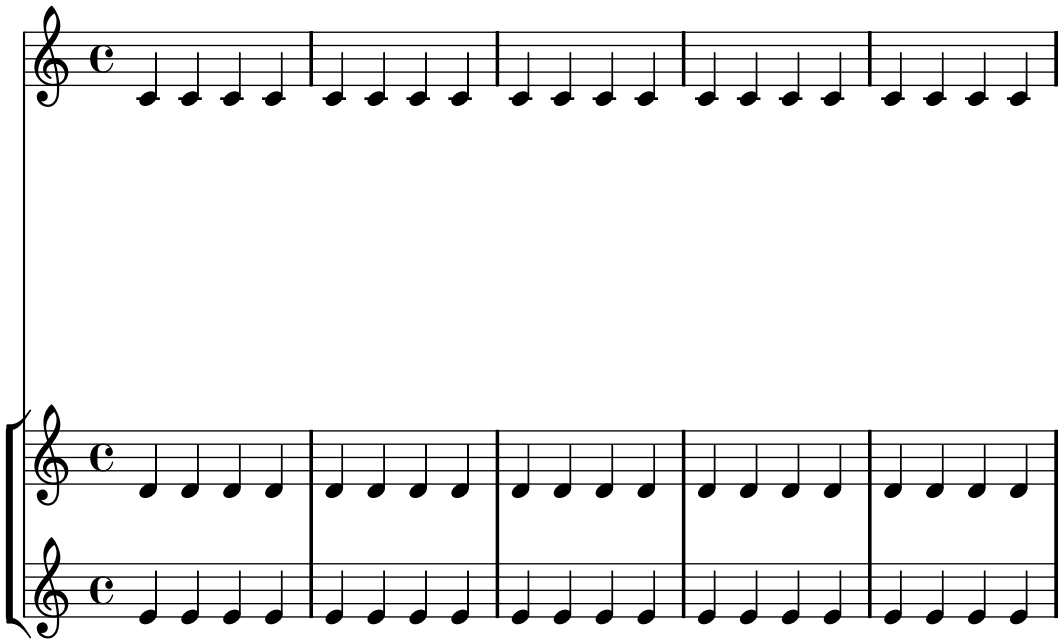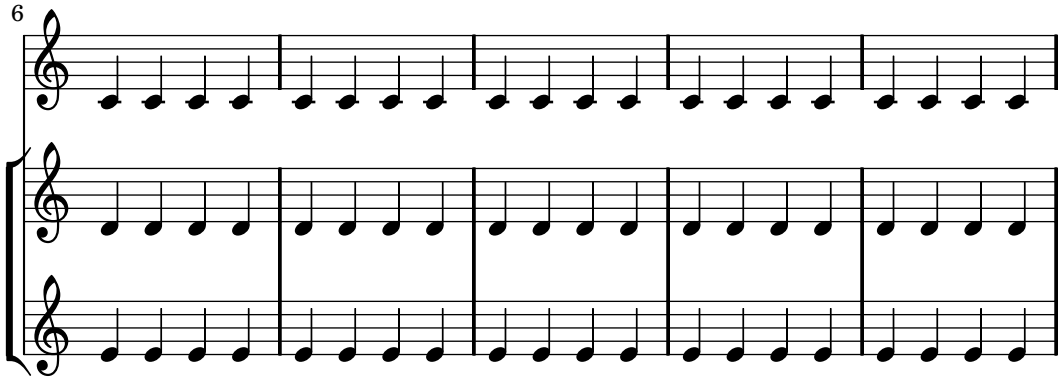


Observe que `line-break-system-details` toma una lista asociativa de una cantidad de valores potencialmente elevada, pero aquí sólo fijamos un valor. Observe también que aquí la propiedad `Y-offset` determina la posición vertical exacta sobre la página en la que se trazará cada uno de los nuevos sistemas.

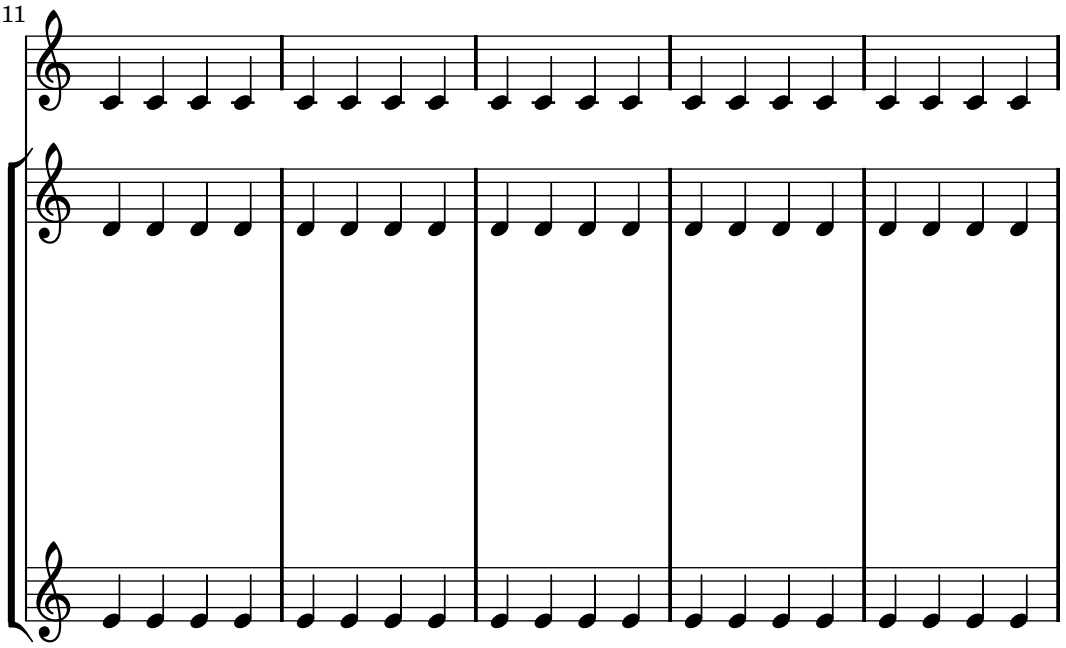
Ahora que hemos establecido explícitamente el punto de origen vertical de cada sistema, podemos también establecer manualmente el punto de origen vertical de cada pentagrama dentro de cada sistema. Lo hacemos usando la subpropiedad `alignment-offsets` de `line-break-system-details`.



Observe que aquí asignamos dos valores distintos al atributo `line-break-system-details` del grob `NonMusicalPaperColumn`. Aunque el atributo alist de `line-break-system-details` acepta muchos parámetros de espaciado adicionales (entre ellos, por ejemplo, un par `X-offset` correspondiente), sólo tenemos que establecer el desplazamiento `Y-offset` y los pares `alignment-offsets` para controlar el punto de origen vertical de cada sistema y pentagrama. Finalmente, observe que `alignment-offsets` especifica el posicionamiento vertical de los pentagramas pero no de los grupos de pentagramas.




6


11

Algunos puntos que tener en cuenta:

- Al usar `alignment-offsets`, la letra cuenta como un pentagrama.
- Las unidades de los números que se pasan a `X-offset`, `Y-offset` y `alignment-offsets` se interpretan como múltiplos de la distancia entre líneas de pentagrama adyacentes. Los valores positivos mueven a los pentagramas y a la letra hacia arriba, los valores negativos mueven los pentagramas y la letra hacia abajo.
- A causa de que los ajustes a `NonMusicalPaperColumn #'line-break-system-details` dados aquí permiten el posicionamiento de pentagramas y sistemas en cualquier lugar de la página, es posible violar los márgenes o los límites del papel, o incluso imprimir pentagramas o sistemas unos encima de otros. Esto se evitaría pasando a los diferentes ajustes unos valores razonables.

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.4.4 Espaciado vertical en dos fases

Nota: El espaciado vertical en dos pasadas está obsoleto y se eliminará en una versión futura de LilyPond. Ahora los sistemas se amplían automáticamente en un solo paso. Véase Sección 4.4.1 [Espaciado vertical dentro de un sistema], página 360.

In order to automatically stretch systems so that they should fill the space left on a page, a two-pass technique can be used:

1. In the first pass, the amount of vertical space used to increase the height of each system is computed and dumped to a file.
2. In the second pass, spacing inside the systems are stretched according to the data in the page layout file.

The `ragged-bottom` property adds space between systems, while the two-pass technique adds space between staves inside a system.

To allow this behavior, a `tweak-key` variable has to be set in each score `\layout` block, and the tweaks included in each score music, using the `\scoreTweak` music function.

```
%% include the generated page layout file:
\includePageLayoutFile

\score {
  \new StaffGroup <<
    \new Staff <<
      %% Include this score tweaks:
      \scoreTweak "scoreA"
      { \clef french c''1 \break c''1 }
    >>
    \new Staff { \clef soprano g'1 g'1 }
    \new Staff { \clef mezzosoprano e'1 e'1 }
    \new Staff { \clef alto g1 g1 }
    \new Staff { \clef bass c1 c1 }
  >>
  \header {
    piece = "Score with tweaks"
  }
```

```
  %% Define how to name the tweaks for this score:
  \layout { #(define tweak-key "scoreA") }
}
```

For the first pass, the `dump-tweaks` option should be set to generate the page layout file.

```
lilypond -dbackend=null -d dump-tweaks <file>.ly
lilypond <file>.ly
```

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

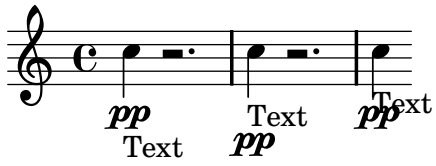
4.4.5 Evitar las colisiones verticales

Podemos decir intuitivamente que algunos objetos de la notación musical pertenecen al pentagrama y otros se sitúan fuera del pentagrama. Entre los objetos que pertenecen al exterior del pentagrama están las marcas de ensayo, las marcas textuales y las indicaciones de dinámica (en lo sucesivo, les llamaremos objetos fuera del pentagrama). La regla de LilyPond para la colocación vertical de los objetos fuera del pentagrama es ponerlos tan cerca del pentagrama como sea posible pero no tan próximos como para que choquen con otro objeto.

LilyPond utiliza la propiedad `outside-staff-priority` para determinar si un grob es un objeto fuera del pentagrama: si `outside-staff-priority` es un número, el grob es un objeto fuera del pentagrama. Además, `outside-staff-priority` informa a LilyPond en qué orden se debe situar los objetos.

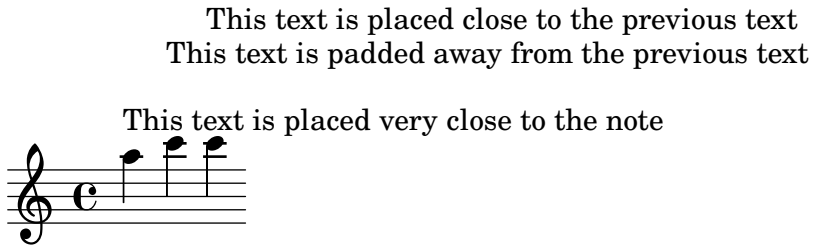
En primer lugar, LilyPond sitúa todos los objetos que no pertenecen al exterior del pentagrama. Después ordena los objetos fuera del pentagrama de acuerdo a su prioridad `outside-staff-priority` (en orden creciente). LilyPond toma los objetos fuera del pentagrama uno a uno y los coloca de forma que no choquen con ningún objeto que ya haya sido colocado. Esto es, si dos grobs fuera del pentagrama compiten por el mismo espacio, el que tiene la prioridad `outside-staff-priority` más baja se colocará más próximo al pentagrama.

```
c4_"Text"\pp
r2.
\once \override TextScript #'outside-staff-priority = #1
c4_"Text"\pp % esta vez el texto estará más cerca del pentagrama
r2.
% ajustando el valor de outside-staff-priority a algo que no sea un número,
% desactivamos la evitación automática de colisiones
\once \override TextScript #'outside-staff-priority = ##f
\once \override DynamicLineSpanner #'outside-staff-priority = ##f
c4_"Text"\pp % ahora se producirá la colisión
```



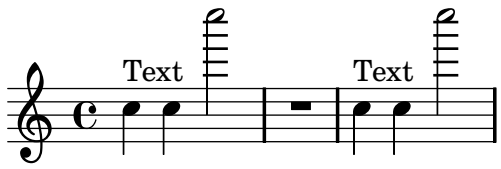
El relleno vertical entre un objeto fuera del pentagrama y los grobs posicionados previamente se puede controlar con `outside-staff-padding`.

```
\once \override TextScript #'outside-staff-padding = #0
a'^"This text is placed very close to the note"
\once \override TextScript #'outside-staff-padding = #3
c^"This text is padded away from the previous text"
c^"This text is placed close to the previous text"
```



De forma predeterminada, los objetos fuera del pentagrama se colocan sólo para evitar una colisión horizontal con los grobs posicionados previamente. Esto puede llevar a situaciones en las que los objetos se colocan muy próximos entre sí en el sentido horizontal. El espacio vertical entre pentagramas se puede fijar también de manera que los objetos fuera del pentagrama se sitúen de forma intercalada. El establecimiento del relleno horizontal `outside-staff-horizontal-padding` ocasiona que un objeto se desplace verticalmente para que tal situación no ocurra.

```
% el marcado está demasiado cerca de la nota siguiente
c4^"Text"
c4
c''2
% ajustar outside-staff-horizontal-padding corrige esto
R1
\once \override TextScript #'outside-staff-horizontal-padding = #1
c,,4^"Text"
c4
c''2
```



Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.
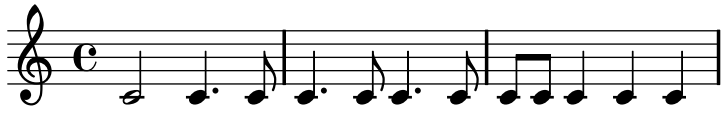
4.5 Espaciado horizontal

4.5.1 Panorámica del espaciado horizontal

El motor de espaciado traduce las diferencias en las duraciones a distancias ampliables (‘muelles’) de distintas longitudes. Las duraciones más largas reciben un espacio mayor y las duraciones más cortas reciben menos. Las duraciones más breves reciben un espacio de tamaño fijo (que se controla mediante `shortest-duration-space` en el objeto Sección “SpacingSpanner” in *Referencia de Funcionamiento Interno*). Cuanto más larga es la duración, más espacio recibe: al doblar una duración se añade un espacio de tamaño fijo (este tamaño se controla mediante `spacing-increment`) a la nota.

Por ejemplo, la pieza siguiente contiene numerosas blancas, negras y corcheas; la corchea va seguida de un ancho de cabeza de blanca (ACB). La negra va seguida de 2 ACB, la blanca por 3 ACB, etc.

```
c2 c4. c8 c4. c8 c4. c8 c8
c8 c4 c4 c4
```



Normalmente, el valor de `spacing-increment` está establecido en 1.2 espacios de pentagrama, lo que es aproximadamente el ancho de una cabeza, y `shortest-duration-space` está establecido en 2.0, lo que significa que la nota más breve recibe 2.4 espacios de pentagrama (2.0 multiplicado por el `spacing-increment`) de espacio horizontal. Este espacio se cuenta a partir del borde izquierdo del símbolo, de manera que las notas más breves van seguidas generalmente por un espacio de 1 ACB.
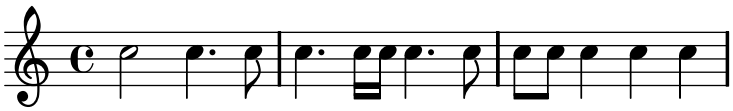
Si siguiésemos el procedimiento anterior exactamente, entonces la adición de una sola fusa a una partitura que usa corcheas y semicorcheas, aumentaría enormemente la anchura de la partitura completa. La nota más breve ya no es la semicorchea, sino la fusa, añadiendo así 1 ACB a cada una de las notas. Para evitarlo, la duración más breve a efectos de espaciado no es la nota más breve de la partitura, sino la que ocurre con más frecuencia.

La duración más breve que es más común se determina de la siguiente manera: en cada compás se determina la duración más breve. La menor duración más común se toma como base para el espaciado, habiendo estipulado que esta duración menor siempre debe ser igual o menor que una corchea. La duración más breve se imprime cuando se ejecuta `lilypond` con la opción `--verbose`.
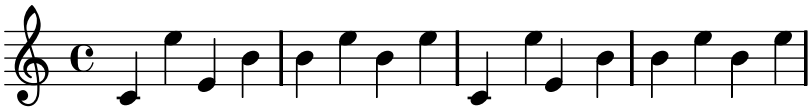
Estas duraciones también se pueden personalizar. Si establecemos la `common-shortest-duration` en Sección “SpacingSpanner” in *Referencia de Funcionamiento Interno*, entonces éste establece la duración base para el espaciado. La duración máxima para esta base (normalmente una corchea), se fija a través de `base-shortest-duration`.

Las notas que son aún más breves que la nota común más breve van seguidas por un espacio proporcional a su duración en relación con la nota común más breve. Así pues, si fuésemos a añadir tan sólo algunas semicorcheas al ejemplo anterior, irían seguidas por medio ACB:

```
c2 c4. c8 c4. c16[ c] c4. c8 c8 c8 c4 c4 c4
```



En la introducción (véase Sección “Engraving” in *Manual de Aprendizaje*), se explicó que las direcciones de las plicas influyen en el espaciado. Esto se controla con la propiedad `stem-spacing-correction` en el objeto Sección “NoteSpacing” in *Referencia de Funcionamiento Interno*. Estos se generan para cada uno de los contextos de Sección “Voice” in *Referencia de Funcionamiento Interno*. El objeto `StaffSpacing` (generado en el contexto de Sección “Staff” in *Referencia de Funcionamiento Interno*) contiene la misma propiedad para controlar el espaciado de las líneas de plica o divisorias. El ejemplo siguiente muestra estas correcciones, una vez con los valores predeterminados y otra con correcciones exageradas:



Está contemplada la notación proporcional; consulte Sección 4.5.5 [Notación proporcional], página 378.

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SpacingSpanner” in *Referencia de Funcionamiento Interno*, Sección “NoteSpacing” in *Referencia de Funcionamiento Interno*, Sección “StaffSpacing” in *Referencia de Funcionamiento Interno*, Sección “SeparationItem” in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No existe ningún mecanismo conveniente para sobreescribir el espaciado manualmente. Se puede usar el siguiente rodeo para insertar espacio adicional en una partitura.

```
\once \override Score.SeparationItem #'padding = #1
```
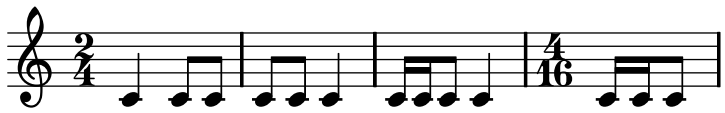
No existe ningún rodeo para disminuir la magnitud de la separación.

4.5.2 Área de espaciado nueva

Se pueden inicar secciones nuevas con diferentes parámetros de espaciado, con `newSpacingSection`. Esto es útil cuando hay secciones que tienen distinta noción de las notas largas y cortas.

En el ejemplo siguiente, el cambio de compás introduce una sección nueva, y por ello las semicorcheas se separan de manera más amplia.

```
\time 2/4
c4 c8 c
c8 c c4 c16[ c c8] c4
\newSpacingSection
\time 4/16
c16[ c c8]
```



La instrucción `\newSpacingSection` crea un nuevo objeto `SpacingSpanner`, y de ahí que se puedan usar nuevas instrucciones de sobreescritura `\override` en dicho punto.

Véase también
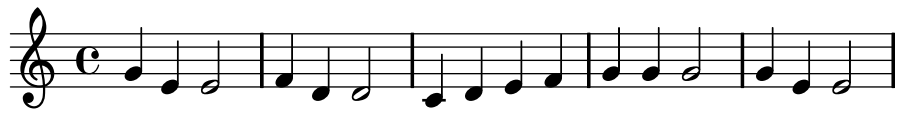
Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

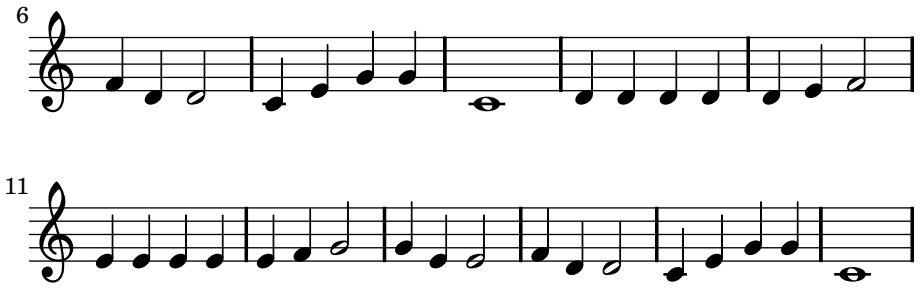
Referencia de funcionamiento interno: Sección “SpacingSpanner” in *Referencia de Funcionamiento Interno*.

4.5.3 Cambiar el espaciado horizontal
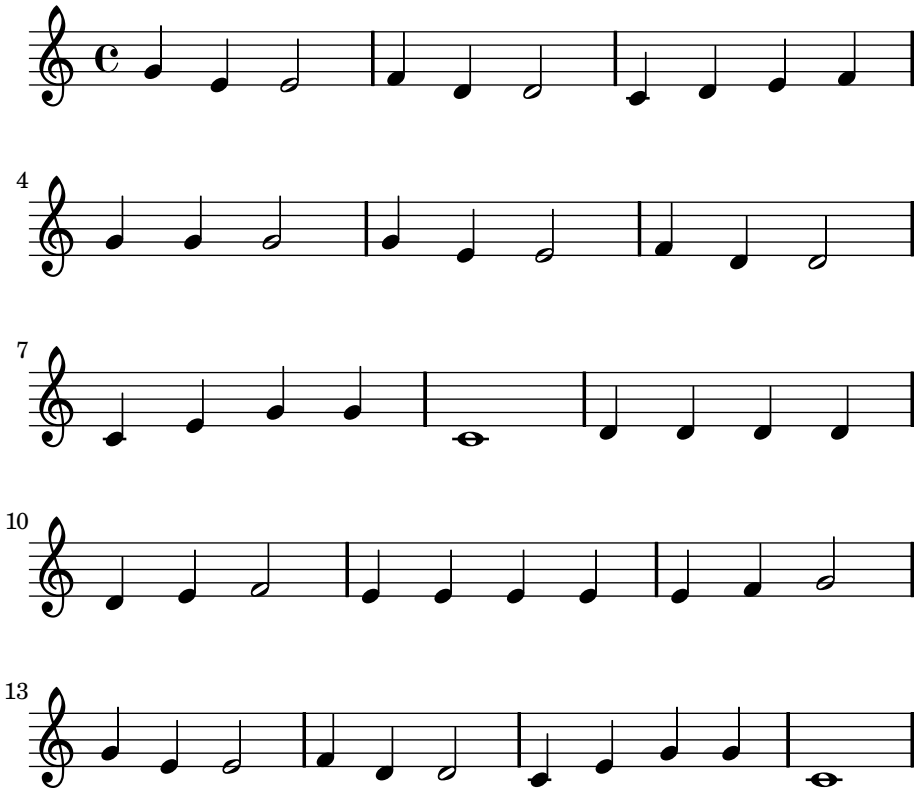
Se puede alterar el espaciado horizontal con la propiedad `base-shortest-duration`. Aquí compararemos la misma música, una vez sin alterar la propiedad, y luego alterándola. Los valores mayores de `ly:make-moment` producen música más pequeña. Observe que `ly:make-moment` construye una duración, por lo que `1 4` es una duración mayor que `1 16`.

```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
}
```





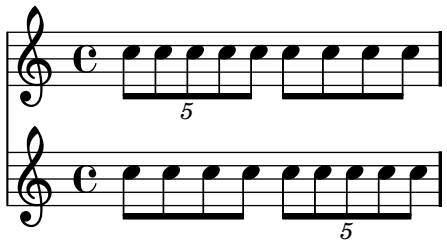

```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner
                #'base-shortest-duration = #(ly:make-moment 1 16)
    }
  }
}
```



Fragmentos de código seleccionados

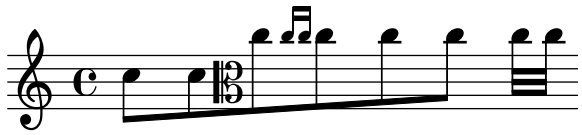
De forma predeterminada, el espaciado en los grupos de valoración especial depende de varios factores ajenos a la duración (como alteraciones, cambios de clave, etc.). Para pasar por alto estos símbolos y forzar un espaciado de duraciones iguales uniforme, use `Score.SpacingSpanner #'uniform-stretching`. Esta propiedad sólo puede cambiarse al principio de la partitura:

```
\new Score \with {
  \override SpacingSpanner #'uniform-stretching = ##t
} <<
  \new Staff{
    \times 4/5 {
      c8 c8 c8 c8 c8
    }
    c8 c8 c8 c8
  }
  \new Staff{
    c8 c8 c8 c8
    \times 4/5 {
      c8 c8 c8 c8 c8
    }
  }
>>
```



Cuando se establece `strict-note-spacing`, las notas se separan sin tener en cuenta las claves, líneas divisorias ni notas de adorno:

```
\override Score.SpacingSpanner #'strict-note-spacing = ##t
\new Staff { c8[ c \clef alto c \grace { c16[ c] } c8 c c]  c32[ c32] }
```



Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.5.4 Longitud de la línea

Los ajustes más básicos que influyen en el espaciado son `indent` y `line-width`. Se definen en el bloque `\layout`. Controlan el sangrado de la primera línea de música y la longitud de las líneas.

Si se fija un valor verdadero para `ragged-right` en el bloque `\layout`, entonces los sistemas terminan en su longitud horizontal natural, en lugar de repartirse horizontalmente hasta llenar toda la línea. Esto es útil para fragmentos cortos, y para comprobar qué tan apretado es el espaciado natural.

La opción `ragged-last` es similar a `ragged-right`, pero afecta sólo a la última línea de la pieza. No se efectúa ninguna restricción sobre dicha línea. El resultado es similar al formateo de los párrafos de texto. En un párrafo, la última línea sencillamente ocupa su longitud horizontal natural.

```
\layout {
  indent = #0
  line-width = #150
  ragged-last = ##t
}
```

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.5.5 Notación proporcional

LilyPond contempla la notación proporcional, un tipo de espaciado horizontal en el que cada nota consume una medida horizontal que equivale exactamente a su duración rítmica. Este tipo de espaciado proporcional es comparable al espaciado horizontal hecho sobre un papel milimetrado. Ciertas partituras del finales del s.XX y principios del s.XXI utilizan notación proporcional para clarificar relaciones rítmicas complejas o para facilitar la colocación de líneas cronométricas u otros gráficos directamente en la partitura.
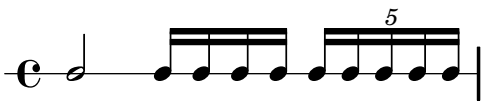
LilyPond contempla cinco ajustes distintos para la notación proporcional, que se pueden usar solos o combinados:

- `proportionalNotationDuration`
- `uniform-stretching`
- `strict-note-spacing`
- `\remove Separating_line_group_engraver`
- `\override PaperColumn #'used = ##t`

En los ejemplos que siguen, exploramos el uso de estos cinco ajustes de la notación proporcional y examinamos la forma en que interactúan.

Comenzamos con el siguiente ejemplo de un solo compás, que usa un espaciado clásico sin justificación por la derecha.

```
\new Score <<
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \times 4/5 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
>>
```



Observe que la blanca que inicia el compás ocupa mucho menos de la mitad de todo el espacio horizontal del compás. De forma similar, las semicorcheas y el cinquillo de semicorcheas con que finaliza el compás ocupan en conjunto mucho más de la mitad de todo el espacio horizontal del compás.

En el grabado clásico, este espaciado puede ser exactamente el que deseamos porque podemos tomar prestado el espacio horizontal de la blanca y conservar el espacio horizontal a lo largo del compás como un todo.

Por otro lado, si quieremos insertar una línea de tiempo graduada o algún otro gráfico encima o debajo de la partitura, necesitamos la notación proporcional. Se activa la notación proporcional con el ajuste `proportionalNotationDuration`.

```
\new Score \with {
  proportionalNotationDuration = #(ly:make-moment 1 20)
} <<
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \times 4/5 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
>>
```



La blanca al princpio del compás y las notas rápidas de la segunda mitad del compás ocupan ahora cantidades iguales de espacio horizontal. Podríamos colocar una línea de tiempo graduada o un gráfico encima o debajo de este ejemplo.

El ajuste `proportionalNotationDuration` es un ajuste de contexto que reside en `Score`. Recordemos que los ajustes de contexto aparecen en uno de tres posibles lugares del archivo de entrada: en un bloque `\with`, en un bloque `\context`, o directamente entre la música precedido por la instrucción `\set`. Como con todos los ajustes de contexto, el usuario puede elegir en cuál de los tres lugares diferentes prefiere establecer el valor de `proportionalNotationDuration`.

El ajuste `proportionalNotationDuration` acepta un solo argumento, que es la duración de referencia contra el que se aplica el espaciado de toda la música. La función Scheme de LilyPond `make-moment` acepta dos argumentos: un numerador y un denominador que, juntos, expresan una cierta fracción de redonda. La llamada `#(ly:make-moment 1 20)`, por tanto, produce una duración de referencia de una nota de un veinteavo de redonda (semicorcheas de cinquillo). Los valores `#(ly:make-moment 1 16)`, `#(ly:make-moment 1 8)` y `#(ly:make-moment 3 97)` son también posibles.

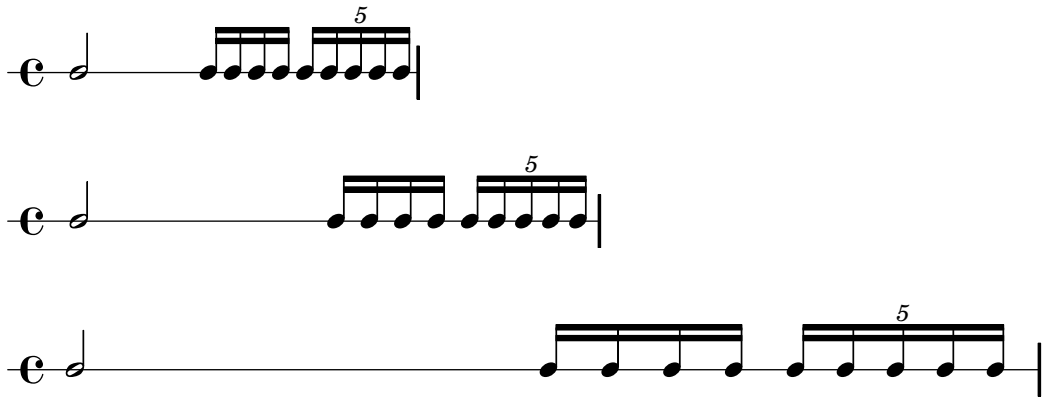
¿Cómo seleccionamos la duración correcta de referencia para pasarla a `proportionalNotationDuration`? Normalmente mediante un proceso de ensayo y error, comenzando con una duración cercana a la más rápida (o más breve) duración de la pieza. Las duraciones de referencia más pequeñas aplican un espaciado más suelto; las duraciones de referencia más largas aplican un espaciado más apretado.

```
\new Score \with {
  proportionalNotationDuration = #(ly:make-moment 1 8)
} <<
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \times 4/5 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
>>
```

```
\new Score \with {
  proportionalNotationDuration = #(ly:make-moment 1 16)
} <<
```

```
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \times 4/5 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
>>

\new Score \with {
  proportionalNotationDuration = #(ly:make-moment 1 32)
} <<
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \times 4/5 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
>>
```



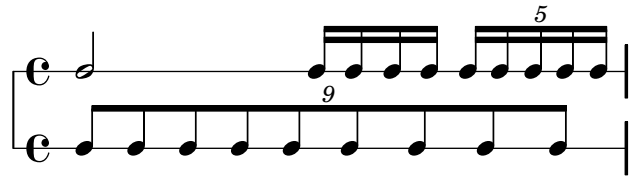
Observe que una duración de referencia demasiado grande (como la corchea, en el ejemplo de arriba) produce un espaciado excesivamente apretado y puede ser causa de colisiones entre las cabezas de nota. Observe también que la notación proporcional en general ocupa más espacio horizontal que el espaciado clásico. El espaciado proporcional aporta claridad rítmica a expensas del espacio horizontal.

Ahora veremos cómo espaciar de forma óptima grupos de valoración especial que se superponen.

Empezamos por examinar qué le ocurre a nuestro ejemplo original, con espaciado clásico, cuando añadimos un segundo pentagrama con un tipo diferente de grupo especial.

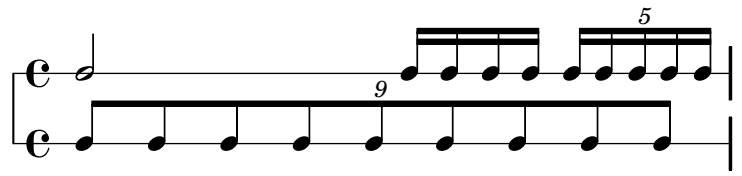
```
\new Score <<
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \times 4/5 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
  \new RhythmicStaff {
```

```
    \times 8/9 {
      c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
    }
  }
>>
```



El espaciado es defectuoso porque las notas regularmente espaciadas del pentagrama inferior no se amplían de manera uniforme. El grabado clásico incluye muy pocos tresillos complejos y así las reglas del grabado clásico pueden generar este tipo de resultado. El establecimiento de `proportionalNotationDuration` remedia esta situación considerablemente.

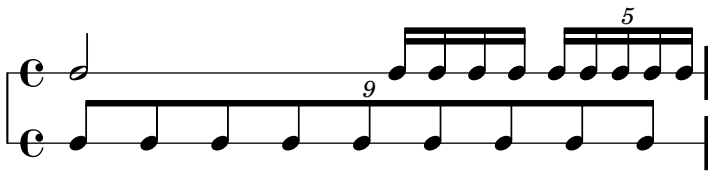
```
\new Score \with {
  proportionalNotationDuration = #(ly:make-moment 1 20)
} <<
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \times 4/5 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
  \new RhythmicStaff {
    \times 8/9 {
      c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
    }
  }
>>
```



Pero si observamos con mucho cuidado podremos ver que las notas de la segunda mitad del 9-illo están espaciadas de forma ligeramente más ancha que las de la primera mitad del 9-illo. Para asegurar una ampliación uniforme, activamos `uniform-stretching`, que es una propiedad de `SpacingSpanner`.

```
\new Score \with {
  proportionalNotationDuration = #(ly:make-moment 1 20)
  \override SpacingSpanner #'uniform-stretching = ##t
} <<
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \times 4/5 {
      c'16 c'16 c'16 c'16 c'16
```

```
    }
  }
  \new RhythmicStaff {
    \times 8/9 {
      c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
    }
  }
>>
```



Nuestro ejemplo de dos pentagramase ahora está espaciado exactamente, nuestras relaciones rítmicas son visualmente claras, y podemos incluir una línea de tiempo graduada o un gráfico, si queremos.

Observe que el paquere de notación proporcional de LilyPond espera que todas las partituras proporcionales establezcan el atributo `uniform-stretching` de `SpacingSpanner` al valor ##t. El establecimiento de `proportionalNotationDuration` sin ajustar también el atributo `uniform-stretching` de `SpacingSpanner` al valor ##t causará, por ejemplo, que los desplazamientos (`skips`) consuman una cantidad de espacio horizontal incorrecta.

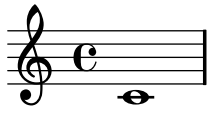
El `SpacingSpanner` es un grob abstracto que reside en el contexto de `Score`. Como con nuestros ajustes de `proportionalNotationDuration`, las sobreescrituras al `SpacingSpanner` sólo pueden ocurrir en uno de estos tres lugares dentro del archivo de entrada: en el bloque `\with`, en el bloque `\context` o directamente dentro de la escritura de notas.
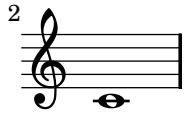
De manera predeterminada, sólo hay un `SpacingSpanner` por `Score`. Esto supone que, por omisión, `uniform-stretching` está activado para la partitura completa o desactivado para la partitura completa. Sin embargo, podemos sobreesacibir este comportamiento y activar distintas posibilidades de espaciado en distintos lugares de la partitura. Lo hacemos con la instrucción `\newSpacingSection`. Consulte Sección 4.5.2 [Área de espaciado nueva], página 375 para más información.

A continuación examinamos los efectos del grabador `Separating_line_group_engraver` y veremos por qué las partituras proporcionales con frecuencia eliminan este grabador. El ejemplo siguiente muestra que hay una pequeña cantidad de espacio “preparatorio” justo antes de la primera nota de cada sistema.

```
\paper {
  indent = #0
}

\new Staff {
  c'1
  \break
  c'1
}
```
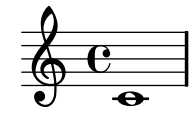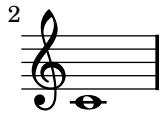




Esta cantidad de espacio preparatorio es la misma ya sea después de una indicación de compás, una armadura o una clave. El grabador `Separating_line_group_engraver` es responsable de este espacio. La eliminación de `Separating_line_group_engraver` reduce este espacio a cero.

```
\paper {
  indent = #0
}

\new Staff \with {
  \remove Separating_line_group_engraver
} {
  c'1
  \break
  c'1
}
```





Los elementos no musicales como la indicación de compás, la armadura, la clave y las alteraciones son problemáticos en notación proporcional. Ninguno de estos elementos tiene duración rítmica. Pero todos ellos consumen espacio horizontal. Las distintas partituras proporcionales abordan este problema de distinta manera.

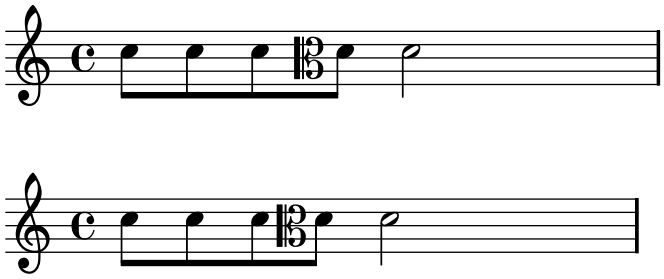
Sería posible evitar los problemas de espaciado con las armaduras, simplemente evitando tenerlas. Esta es una opción válida pues casi todas las partituras proporcionales son música contemporánea. Lo mismo puede valer para las indicaciones de compás, especialmente para las partituras que incluyen una línea de tiempo graduada u otro gráfico. Pero estas partituras son excepcionales y casi todas las partituras proporcionales incluyen al menos unas pocas indicaciones de compás. Las claves y las alteraciones son aún más esenciales.

Así pues ¿qué estrategias existen para el espaciado de los elementos no musicales en un contexto proporcional? Una buena opción es la propiedad `strict-note-spacing` de `SpacingSpanner`. Compare las dos partituras siguientes:

```
\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1 16)
  c''8
  c''8
  c''8
  \clef alto
  d'8
  d'2
}

\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1 16)
```

```
  \override Score.SpacingSpanner #'strict-note-spacing = ##t
  c''8
  c''8
  c''8
  \clef alto
  d'8
  d'2
}
```



Las dos poartituras son proporcionales, pero el espaciado de la primera es muy suelto a causa del cambio de clave. Sin embargo, el espaciado de la segunda partitura se mantiene estricto, porque `strict-note-spacing` está activado. La activación de `strict-note-spacing` hace que el ancho de las indicaciones de compás, armaduras, cambios de clave y alteraciones no tomen parte en el algoritmo de espaciado.

Además de los ajustes dados aquí, hay otros que aparecen con frecuencia en las partituras proporcionales. Entre ellos están:

- `\override SpacingSpanner #'strict-grace-spacing = ##t`
- `tupletFullLength = ##t`
- `\override Beam #'breakable = ##t`
- `\override Glissando #'breakable = ##t`
- `\override TextSpanner #'breakable = ##t`
- `\remove Forbid_line_break_engraver in the Voice context`

Estos ajustes dan a las notas de adorno un espaciado estricto, extienden los corchetes de grupo especial para que marquen tanto los puntos de comienzo como de final, y permiten que los elementos de extensión se dividan entre los sistemas y las páginas. Consulte las partes respectivas del manual para ver estos ajustes relacionados.

Véase también

Referencia de la notación: Sección 4.5.2 [Área de espaciado nueva], página 375.

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.6 Encajar la música en menos páginas

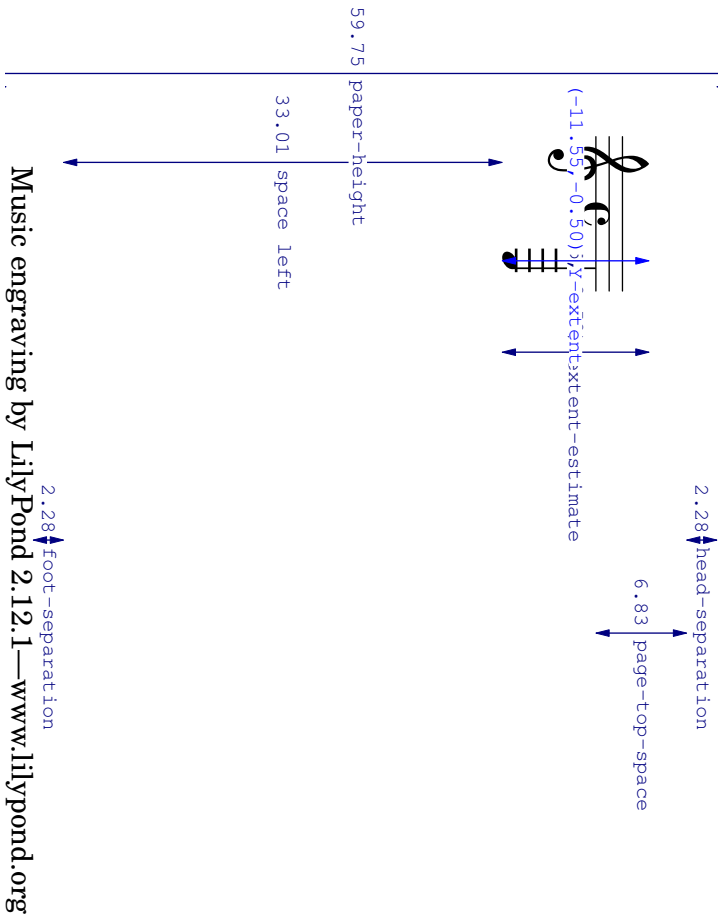
En ocasiones, podemos terminar con uno o dos pentagramas en una segunda página (o tercera, o cuarta...). Es fastidioso, especialmente cuando vemos que las páginas anteriores parecen tener espacio de sobra.

Al investigar los problemas de disposición, una herramienta de valor incalculable es `annotate-spacing`. Esta instrucción imprime los valores de un cierto número de variables de espaciado; para ver más detalles consulte la sección siguiente, Sección 4.6.1 [Mostrar el espaciado], página 385.

4.6.1 Mostrar el espaciado

Para presentar gráficamente las dimensiones de las variables de disposición vertical que pueden verse alteradas por el formato de la página, establezca `annotate-spacing` en el bloque `\paper`:

```
#(set-default-paper-size "a6" 'landscape)
\book {
  \score { { c4 } }
  \paper { annotate-spacing = ##t }
}
```



Todas las dimensiones de disposición se muestran en espacios de pentagrama, independientemenre de las unidades especificadas en los bloques `\paper` o `\layout`. Por ejemplo, `paper-height` tiene un valor de 59.75 espacios de pentagrama, usando el tamaño predeterminado de pentagrama de 20 puntos, que equivale a 148 milímetros, la altura de una hoja de papel `a6` en orientación apaisada. Las parejas (a,b) son intervalos, donde a es el límite inferior y b es el límite superior del intervalo.

Véase también

Fragmentos de código: Sección “Spacing” in *Fragmentos de código*.

4.6.2 Cambiar el espaciado

La salida de `annotate-spacing` revela las dimensiones verticales con gran detalle. Para ver más detalles acerda de la modificación de los márgenes y otras variables de diseño de la página, consulte Sección 4.1.2 [Formateo de las páginas], página 346.

Aparte de los márgenes, hay algunas otras opciones para ahorrar espacio:

- Forzar los sistemas para que se junten lo más posible entre sí (de forma que quepan la mayor cantidad de ellos en la página) y al mismo tiempo estén espaciados de forma que no exista un espacio vacío en la parte baja del papel.
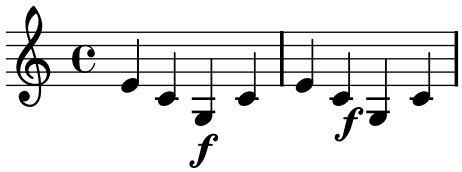
```
\paper {
  between-system-padding = #0.1
  between-system-space = #0.1
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Forzar el número de sistemas. Por ejemplo, si la disposición predeterminada tiene 11 sistemas, la asignación siguiente fuerza una disposición con 10 sistemas.

```
\paper {
  system-count = #10
}
```
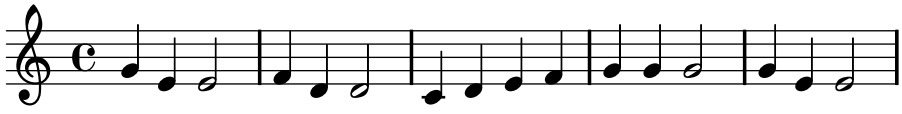
- Evitar (o reducir en número) los objetos que aumentan el tamaño vertical de un sistema. Por ejemplo, las repeticiones de primera y segunda vez (o repeticiones con finales alternativos) necesitan espacio adicional. Si estas repeticiones se reparten a lo largo de dos sistemas, ocupan más espacio que un sistema con las casillas de repetición y otro sistema sin ellas. Por ejemplo, las indicaciones dinámicas que se 'salen' de un sistema se pueden acercar al pentagrama:

```
e4 c g\f c
\override DynamicText #'extra-offset = #'( -2.2 . 2.0)
e4 c g\f c
```
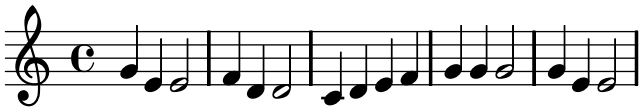


- Alterar el espaciado horizontal por medio de `SpacingSpanner`. Para ver más detalles, consulte Sección 4.5.3 [Cambiar el espaciado horizontal], página 375. El ejemplo siguiente ilustra el espaciado predeterminado:

```
\score {
  \relative c'' {
    g4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
}
```



El ejemplo siguiente modifica `common-shortest-duration` de un valor de `1/4` (negra) a `1/2` (blanca). La negra es la nota más común y la más breve del ejemplo, por lo que al agrandar esta duración se produce un efecto de 'apretujamiento':

```
\score {
  \relative c'' {
    g4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner
        #'common-shortest-duration = #(ly:make-moment 1 2)
    }
  }
}
```



La propiedad `common-shortest-duration` no se puede modificar de manera dinámica, por lo que se debe situar siemmpre dentro de un bloque `\context` de forma que se aplique a la partitura completa.

Véase también

Referencia de la notación: Sección 4.1.2 [Formateo de las páginas], página 346, Sección 4.5.3 [Cambiar el espaciado horizontal], página 375.

Fragmentos de código: Sección "Spacing" in *Fragmentos de código*.

5 Cambiar los valores por omisión

El objetivo del diseño de LilyPond es proporcionar la más alta calidad de los resultados, de forma predeterminada. A pesar de ello, podría tener que cambiar este resultado predeterminado. La disposición sobre el papel se controla a través de un amplio número de 'botones e interruptores' llamados en su conjunto 'propiedades'. En el Manual de aprendizaje podemos encontrar una introducción en forma de tutorial al acceso y modificación de estas propiedades, véase Sección "Tweaking output" in *Manual de Aprendizaje*. Éste debería leerse en primer lugar. Este capítulo cubre un terreno similar, pero con un estilo más adecuado para un manual de referencia.

La descripción definitiva de los controles que están dipsonibles para su ajuste fino están en un documento aparte: Sección "la Referencia de funcionamiento interno" in *Referencia de Funcionamiento Interno*. Dicho manual relaciona todas las variables, funciones y opciones que se encuentran disponibles en LilyPond. Está escrito como un documento HTML, que se puede encontrar en on-line, pero que también va incluido en el paquete de la documentación de LilyPond.

Internamente, LilyPond utiliza el lenguaje Scheme (un dialecto de LISP) para aportar la infraestructura. La sobreescritura de las decisiones de disposición da acceso efectivo a las interioridades del programa, lo que requiere código de Scheme como entrada. Los elementos de Scheme se inauguran dentro de un archivo `.ly` con el símbolo de cuadradillo `#`.[1]

5.1 Contextos de interpretación

Esta sección explica qué son los contextos y cómo modificarlos.

Véase también

Manual de aprendizaje: Sección "Contexts and engravers" in *Manual de Aprendizaje*.

Archivos de inicio: '`ly/engraver-init.ly`', '`ly/performer-init.ly`'.

Fragmentos de código: Sección "Contexts and engravers" in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Contexts" in *Referencia de Funcionamiento Interno*, Sección "Engravers and Performers" in *Referencia de Funcionamiento Interno*.

5.1.1 Explicación de los contextos

Los contextos se disponen de forma jerárquica:

Score: el contexto maestro

Este es el contexto de notación del nivel más alto. Ningún otro contexto puede contener a un contexto Score. De forma predeterminada, el contexto Score maneja la administración de las indicaciones de compás y se asegura de que ciertos elementos como claves, compases y armaduras están siempre alineados entre los distintos pentagramas.

Se crea implícitamente una instancia del contexto Score cuando se procesa un bloque `\score {...}` o `\layout {...}`, o explícitamente cuando se ejecuta una instrucción `\new Score`.

Contextos del nivel superior: contenedores de pentagramas

StaffGroup

Agrupa pentagramas y añade un corchete en la parte izquierda, formando un grupo. Las líneas divisorias de los pentagramas contenidos se conectan verticalmente. `StaffGroup` sólo consiste en una colección de pentagramas, con un corchete delante y líneas divisorias de arriba a abajo.

ChoirStaff

[1] Sección "Scheme tutorial" in *Manual de Aprendizaje* contiene un breve tutorial sobre la introducción de números, listas, cadenas y símbolos en Scheme.

Idéntico a `StaffGroup` excepto que las barras de compás de los pentagramas contenidos no se conectan verticalmente.

GrandStaff

Un grupo de pentagramas, con una llave en la parte izquierda que abarca el grupo. Las barras de compás de los pentagramas contenidos se conectan verticalmente.

PianoStaff

Igual que `GrandStaff`, pero contempla la posibilidad de poner el nombre del instrumento a la izquierda del sistema.

Contextos de nivel intermedio: pentagramas

Staff

Maneja claves, barras de compás, tonalidades y alteraciones accidentales. Puede contener contextos de `Voice`.

RhythmicStaff

Como `Staff`, pero para imprimir ritmos. Se ignoran las alturas de las notas; las notas se imprimen sobre una línea.

TabStaff

Contexto para generar tablaturas. De forma predeterminada dispone la expresión musical como una tablatura de guitarra, impresa sobre seis líneas.

DrumStaff

Maneja el tipografiado para instrumentos de percusión. Puede contener contextos `DrumVoice`.

VaticanaStaff

Igual que `Staff`, excepto que está pensado para tipografiar piezas en estilo gregoriano.

MensuralStaff

Igual que `Staff`, excepto que está diseñado para tipografiar piezas en estilo mensural.

Contextos del nivel más bajo: voces

Los contextos del mismo nivel que Voice dan un valor inicial a ciertas propiedades e inician los grabadores correspondientes. Siendo contextos del nivel más bajo, no pueden contener a otros contextos.

Voice

Corresponde a una voz sobre un pentagrama. este contexto maneja la conversión de las indicaciones dinámicas, plicas, barras, subíndices y superíndices, ligaduras de expresión y de unión, y silencios. Tenemos que crear instancias explícitas de este contexto si necesitamos varias voces en el mismo pentagrama.

VaticanaVoice

Lo mismo que `Voice`, excepto que está diseñado para tipografiar piezas en estilo gregoriano.

MensuralVoice

Lo mismo que `Voice`, con modificaciones para el tipografiado de piezas en estilo mensural.

Lyrics

Corresponde a una voz con letra. Maneja la impresión de una sola línea de letra.

DrumVoice

El contexto de voz utilizado en una pauta de percusión.

FiguredBass

El contexto en que los objetos `BassFigure` se crean a partir de la entrada escrita en el modo `\figuremode`.

TabVoice

El contexto de voz utilizado dentro de un contexto `TabStaff`. Se suele dejar que se cree implícitamente.

ChordNames

Tipografía nombres de acordes.

5.1.2 Crear contextos

Para partituras que sólo tienen una voz y un pentagrama, los contextos se crean automáticamente. Para partituras más complejas, es necesario crearlos a mano. Existen tres instrucciones que hacen esto.
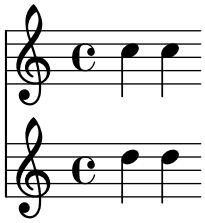
- La instrucción más fácil es `\new`, y es también la más rápida de escribir. Se antepone a una expresión musical, por ejemplo

  `\new` *`tipo expresión_musical`*

  donde *tipo* es el nombre de un contexto (como `Staff` o `Voice`). Esta instrucción crea un contexto nuevo, y empieza a interpretar la *expresión_musical* con él.

  Una aplicación práctica de `\new` es una partitura con muchos pentagramas. Cada parte que debe ir en su propio pentagrama, va precedida de `\new Staff`.

  ```
  <<
    \new Staff { c4 c }
    \new Staff { d4 d }
  >>
  ```

  La instrucción `\new` puede también dar nombre al contexto,

  `\new` *`tipo = identificador música`*

  Sin embargo, este nombre especificado por el usuario sólo se utiliza si no hay ya otro contexto anterior con el mismo nombre.

- Como `\new`, la instrucción `\context` también dirige una expresión musical a un objeto de contexto, pero da al contexto un nombre explícito. La sintaxis es

  `\context` *`tipo = identificador música`*

  En esta forma, la instrucción buscará un contexto existente del *tipo* especificado que tenga el nombre *identificador*. Si ese contexto aún no existe, se crea un contexto nuevo con el nombre especificado. Esto es útil si nos vamos a referir más tarde al contexto. Por ejemplo, cuando se escribe la letra, la melodía está dentro de un contexto con nombre

  `\context Voice = "tenor"` *`música`*

  de forma que los textos se puedan alienar correctamente con sus notas,
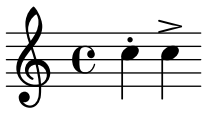
  `\new Lyrics \lyricsto "tenor"` *`letra`*

  Otro uso posible de los contextos con nombre es la fusión de dos expresiones musicales distintas en un solo contexto. En el siguiente ejemplo, se introducen por separado las articulaciones y las notas,

```
musica = { c4 c4 }
decoracion = { s4-. s4-> }
```

se combinan enviando los dos al mismo contexto `Voice`,

```
<<
  \new Staff \context Voice = "A" \musica
  \context Voice = "A" \decoracion
>>
```



Con este mecanismo, es posible definir un Urtext (una edición original), con la posibilidad de poner articulaciones distintas sobre las mismas notas.

- La tercera instrucción para crear contextos es

  `\context` *`tipo música`*

  Esto es similar a `\context` con `=` *`identificador`*, pero se corresponde con cualquier contexto del tipo *tipo*, sin importar qué nombre se le ha dado.

  Esta variante se usa con expresiones musicales que se pueden interpretar en varios niveles. Por ejemplo, la instrucción `\applyOutput` (véase Sección 6.5.2 [Ejecutar una función sobre todos los objetos de la presentación], página 447). Sin una instrucción `\context` explícita, normalmente se aplicaría a `Voice`

  ```
  \applyOutput #'contexto #función   % aplicar al contexto Voice
  ```

  Para que se interprete dentro de los niveles de `Score` o `Staff`, utilice las siguientes formas:

  ```
  \applyOutput #'Score #función
  \applyOutput #'Staff #función
  ```

5.1.3 Modificar los complementos (plug-ins) de contexto

Los contextos de notación (como `Score` y `Staff`) no sólo almacenan propiedades, también contienen «plug-ins» o complementos llamados 'grabadores' que crean elementos de notación. Por ejemplo, el contexto `Voice` contiene un grabador `Note_head_engraver` que crea las cabezas de nota y el contexto `Staff` contiene un grabador `Key_signature_engraver` que crea la indicación de compás.
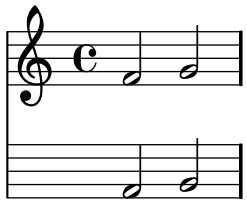
Para ver una descripción completa de todos y cada uno de los complementos, consulte Referencia de funcionamiento interno ↦ Traducción ↦ Grabadores. Cada contexto que se describe en Referencia de funcionamiento interno ↦ Traducción ↦ Contexto. relaciona los grabadores que se usan para ese contexto.

Puede ser de utilidad jugar un poco con estos complementos. Se hace iniciando un contexto nuevo con `\new` o `\context` y modificándolo:

```
\new contexto \with {
  \consists ...
  \consists ...
  \remove ...
  \remove ...
  etc.
}
{
  ..música..
}
```

donde los . . . debe ser el nombre de un grabador. Aquí tenemos un ejemplo sencillo que suprime los grabadores `Time_signature_engraver` y `Clef_engraver` de un contexto `Staff`:

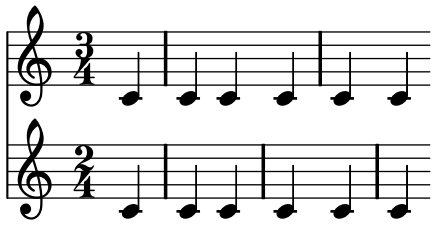
```
<<
  \new Staff {
    f2 g
  }
  \new Staff \with {
     \remove "Time_signature_engraver"
     \remove "Clef_engraver"
  } {
    f2 g2
  }
>>
```



En el segundo pentagrama no hay indicación de compás ni clave. Éste es un método bastante rudimentario de hacer que desaparezcan los objetos porque afecta a todo el pentagrama. Este método también afecta al espaciado, lo que puede ser deseable o no serlo. Se muestran métodos más sofisticados para quitar objetos en Sección “Visibility and color of objects” in *Manual de Aprendizaje*.

El ejemplo siguiente muestra una aplicación práctica. Normalmente las líneas divisorias y las indicaciones de compás están sincronizadas a lo largo de toda la partitura. Lo hacen los grabadores `Timing_translator` y `Default_bar_line_engraver`. Estos complementos mantienen al día la administración de las indicaciones de compás, posición dentro del compás, etc. Moviendo estos grabadores desde el contexto de `Score` al de `Staff`, podemos conseguir una partitura en la que cada pentagrama tiene su propio compás independiente.

```
\new Score \with {
  \remove "Timing_translator"
  \remove "Default_bar_line_engraver"
} <<
  \new Staff \with {
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  } {
      \time 3/4
      c4 c c c c c
  }
  \new Staff \with {
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  } {
       \time 2/4
       c4 c c c c c
  }
>>
```

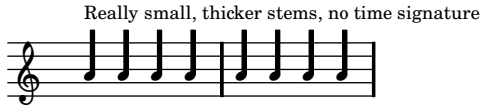


5.1.4 Cambiar los valores por omisión de los contextos

Los ajustes de contexto que están preparados para usarse de forma predeterminada en los contextos `Score`, `Staff` y `Voice`, se pueden especificar dentro de un bloque `\layout`, como se ilustra en el ejemplo siguiente. El bloque `\layout` se debe colocar dentro del bloque `\score` en que se quiere que haga efecto, pero fuera de la música.

Observe que la propia instrucción `\set` y el contexto se deben omitir cuando se especifican de esta manera los valores de contexto predeterminados:

```
\score {
  \relative c'' {
    a4^"Really small, thicker stems, no time signature" a a a
    a a a a
  }
  \layout {
    \context {
      \Staff
      fontSize = #-4
      \override Stem #'thickness = #4.0
      \remove "Time_signature_engraver"
    }
  }
}
```



En este ejemplo, la instrucción `\Staff` especifica que los ajustes siguientes se apliquen a todos los pentagramas dentro del bloque de partitura.

Se pueden realizar de forma similar modificaciones al contexto `Score` o a todos los contextos `Voice`.

Advertencias y problemas conocidos

No es posible recolectar cambios de contexto dentro de una variable y aplicarlos a una definición de `\context` por referencia a dicha variable.

La instrucción `\RemoveEmptyStaffContext` sobreescribe nuestros ajustes en curso para `\Staff`. Si queremos cambiar los valores predeterminados para un pentagrama que utilice `\RemoveEmptyStaffContext`, debe hacerlo después de llamar a `\RemoveEmptyStaffContext`, o sea

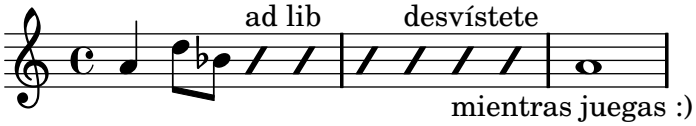
```
\layout {
  \context {
    \RemoveEmptyStaffContext

    \override Stem #'thickness = #4.0
  }
}
```

5.1.5 Definir contextos nuevos

Los contextos específicos, como `Staff` y `Voice`, están construidos a base de bloques sencillos. Es posible crear nuevos tipos de contextos con combinaciones distintas de añadidos grabadores.

El siguiente ejemplo muestra cómo construir un tipo diferente de contexto de `Voice` partiendo de cero. Será parecido a `Voice`, pero imprime solamente cabezas centradas en forma de barra inclinada. Se puede usar para indicar improvisación en piezas de jazz,



Estos ajustes se definen dentro de un bloque `\context` que a su vez está dentro de un bloque `\layout`,

```
\layout {
  \context {
    ...
  }
}
```

En el siguiente análisis, la entrada de ejemplo que se muestra debe ir en el lugar de los puntos suspensivos ... del fragmento anterior.

En primer lugar es necesario definir un nombre para el nuevo contexto:

```
\name ImproVoice
```

Debido a que es parecido al contexto `Voice`, queremos órdenes que funcionen sobre contextos de `Voice` (existentes) para que siga funcionando. Esto se consigue dando al contexto nuevo un alias `Voice`,

```
\alias Voice
```

El contexto imprimirá notas y textos explicativos, por ello tenemos que añadir los grabadores que aportan esta funcionalidad,

```
\consists Note_heads_engraver
\consists Text_engraver
```

Pero sólo necesitamos esto en la línea central,

```
\consists Pitch_squash_engraver
squashedPosition = #0
```

El grabador Sección “Pitch_squash_engraver” in *Referencia de Funcionamiento Interno* modifica las cabezas de nota (creadas por el grabador Sección “Note_heads_engraver” in *Referencia de Funcionamiento Interno*) y establece sus posiciones verticales al valor de `squashedPosition`, en este caso `0`, la línea central.

Las notas parecen barras inclinadas y no tienen plica,

```
\override NoteHead #'style = #'slash
\override Stem #'transparent = ##t
```

Todos estos añadidos tienen que cooperar, y esto se consigue con un añadido especial, que se debe marcar con la palabra clave `\type`. Este será siempre `Engraver_group`,

```
\type "Engraver_group"
```

Al juntarlo todo, obtenemos

```
\context {
  \name ImproVoice
  \type "Engraver_group"
```

```
  \consists "Note_heads_engraver"
  \consists "Text_engraver"
  \consists Pitch_squash_engraver
  squashedPosition = #0
  \override NoteHead #'style = #'slash
  \override Stem #'transparent = ##t
  \alias Voice
}
```

Los contextos dan lugar a jerarquías. Queremos colgar el contexto `ImproVoice` bajo el contexto `Staff`, como simples `Voice`s normales. Por tanto, modificamos la definición de `Staff` con la instrucción `\accepts` (acepta),

```
\context {
  \Staff
  \accepts ImproVoice
}
```

Lo opuesto a `\accepts` (acepta) es `\denies` (deniega), lo que a veces se necesita cuando se están reutilizando definiciones de contexto existentes.

Ponemos ambos dentro de un bloque `\layout`, como
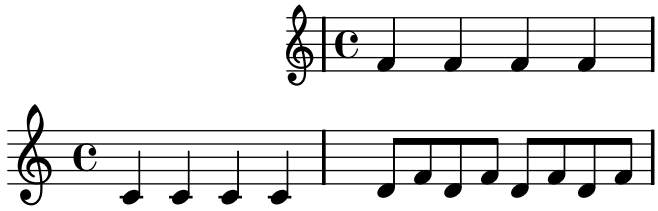
```
\layout {
  \context {
    \name ImproVoice
    ...
  }
  \context {
    \Staff
    \accepts "ImproVoice"
  }
}
```

Así pues, la salida que aparece al comienzo de esta sub-sección se puede escribir como

```
\relative c'' {
  a4 d8 bes8
  \new ImproVoice {
    c4^"ad lib" c
    c4 c^"undress"
    c c_"while playing :)"
  }
  a1
}
```

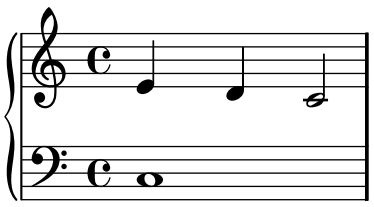
5.1.6 Alinear los contextos

Los contextos nuevos se pueden alinear por encima o por debajo de otros contextos existentes. Esto podría ser de utilidad al preparar un pentagrama vocal (Sección “Vocal ensembles” in *Manual de Aprendizaje*) y un ossia,
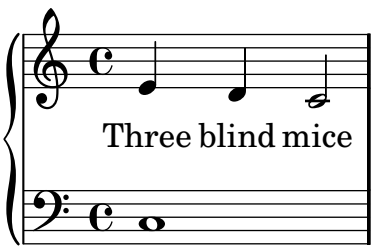


Los contextos como `PianoStaff` pueden llevar dentro otros contextos anidados. Los contextos que se pueden aceptar para su anidamiento están definidos por la lista "accepts" (acepta) de un contexto. Los contextos que no están en esta lista se colocan debajo del contexto exterior en la partitura impresa. Por ejemplo, el contexto `PianoStaff` está definido para que acepte contextos `Staff` y `FiguredBass` de forma predeterminada, pero no un contexto `Lyrics`, por ejemplo. Así pues, en la siguiente estructura la letra se sitúa debajo del sistema de piano en lugar de colocarse entre los dos pentagramas:

```
\new PianoStaff
<<
  \new Staff { e4 d c2 }
  \addlyrics { Three blind mice }
  \new Staff {
    \clef "bass"
    { c,1 }
  }
>>
```



La lista "accepts" de un contexto se puede modificar para que incluya contextos anidados adicionales, y así si quisiéramos que la letra apareciese entre los dos pentagramas podríamos usar:

```
\new PianoStaff \with { \accepts Lyrics }
<<
  \new Staff { e4 d c2 }
  \addlyrics { Three blind mice }
  \new Staff {
    \clef "bass"
    { c,1 }
  }
>>
```
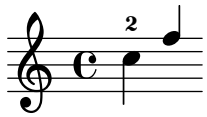


Lo contrario de `\accepts` (acepta) es `\denies` (deniega); esto suprime un contexto de la lista "accepts".

5.2 Explicación del Manual de referencia de funcionamiento interno

5.2.1 Navegar por la referencia del programa

Supongamos que queremos mover la indicación de digitación del fragmento siguiente:

```
c-2
\stemUp
f
```



Si hace una visita a la documentación en busca de instrucciones de digitación (en [Indicaciones de digitación], página 155), encontrará:

Véase también

Referencia de funcionamiento interno: Sección “Fingering” in *Referencia de Funcionamiento Interno*.

La referencia del programador se encuentra disponible en forma de documento HTML. Se recomienda mucho que lo lea en la forma HTML, bien en línea o bien descargando los archivos de la documentación HTML. Esta sección sería mucho más difícil de entender si está utilizando el manual en formato PDF.

Siga el enlace que lleva a Sección “Fingering” in *Referencia de Funcionamiento Interno*. Al principio de la página, puede ver

Los objetos de digitación se crean por parte de: Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno* y Sección “New_fingering_engraver” in *Referencia de Funcionamiento Interno*.

Siguiendo los enlaces relacionados dentro de la referencia del programa, podemos seguir el flujo de información dentro del programa:

- Sección “Fingering” in *Referencia de Funcionamiento Interno*: los objetos Sección “Fingering” in *Referencia de Funcionamiento Interno* se crean por parte de: Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno*
- Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno*: Tipos de música aceptados: Sección “fingering-event” in *Referencia de Funcionamiento Interno*
- Sección “fingering-event” in *Referencia de Funcionamiento Interno*: El tipo de evento musical `fingering-event` está descrito en Expresiones musicales con el nombre de Sección “FingeringEvent” in *Referencia de Funcionamiento Interno*

Este camino se recorre en contra de la corriente de información del programa: comienza por la salida y acaba en el evento de entrada. También podríamos haber empezado por un evento de la entrada, y leído siguiendo el flujo de información terminando en su caso en el objeto (u objetos) de la salida.

La referencia del programa también se puede examinar como un documento normal. Contiene capítulos que tratan de `Music definitions` de la Sección “Translation” in *Referencia de Funcionamiento Interno*, y del Sección “Backend” in *Referencia de Funcionamiento Interno*. Cada uno de los capítulos relaciona todas las definiciones utilizadas y todas las propiedades que se pueden ajustar.

5.2.2 Interfaces de la presentación

La página HTML que pudimos ver en la sección anterior describe el objeto de presentación llamado Sección “Fingering” in *Referencia de Funcionamiento Interno*. Dicho objeto es un símbolo dentro de la partitura. Tiene propiedades que guardan números (como grosores y direcciones), pero también punteros a objetos relacionados. Un objeto de presentación también se llama un *Grob*, que es una abreviatura de Graphical Object (objeto gráfico). Para ver más detalles acerca de los objetos gráficos o Grobs, consulte Sección “grob-interface” in *Referencia de Funcionamiento Interno*.

La página dedicada a `Fingering` relaciona las definiciones del objeto `Fingering`. Por ejemplo, la página dice

`relleno` (dimensión, en espacios de pentagrama):

`0.5`

lo que significa que el número se mantendrá a una distancia de al menos 0.5 de la cabeza de la nota.

Cada objeto de presentación puede tener varias funciones como elemento notacional o tipográfico. Por ejemplo, el objeto de digitación Fingering tiene los siguientes aspectos

- Su tamaño es independiente del espaciado horizontal, al contrario de las ligaduras o las barras de las figuras.
- Es un elemento de texto. Casi seguro que es un texto muy corto.
- este elemento de texto se tipografía con un tipo de letra, no como las ligaduras o las barras de las figuras.
- Horizontalmente, el centro del símbolo se debe alinear con el centro de la cabeza de la nota.
- Verticalmente, el símbolo se coloca cerca de la nota y del pentagrama.
- La posición vertical también está coordinada con otros símbolos de superíndice y de subíndice.

Cada uno de estos aspectos se capta en lo que se llaman *interfaces*, que se relacionan al final de la página dedicada a Sección “Fingering” in *Referencia de Funcionamiento Interno*

Este objeto contempla los siguientes interfaces: Sección “item-interface” in *Referencia de Funcionamiento Interno*, Sección “self-alignment-interface” in *Referencia de Funcionamiento Interno*, Sección “side-position-interface” in *Referencia de Funcionamiento Interno*, Sección “text-interface” in *Referencia de Funcionamiento Interno*, Sección “text-script-interface” in *Referencia de Funcionamiento Interno*, Sección “font-interface” in *Referencia de Funcionamiento Interno*, Sección “finger-interface” in *Referencia de Funcionamiento Interno* y Sección “grob-interface” in *Referencia de Funcionamiento Interno*.

Al pulsar sobre cualquiera de los enlaces nos desplazaremos a la página del respectivo interfaz del objeto. Cada interfaz tiene un cierto número de propiedades. Algunas de ellas no son para que el usuario las pueda ajustar (‘Propiedades internas’), pero otras sí se pueden modificar.

Hemos estado hablando de *el* objeto `Fingering`, pero realmente esto no significa mucho. El archivo de inicialización (véase Sección 3.1.3 [Estructura del archivo], página 318) ‘`scm/define-grobs.scm`’ muestra el alma del ‘objeto’,

```
(Fingering
  . ((padding . 0.5)
     (avoid-slur . around)
     (slur-padding . 0.2)
     (staff-padding . 0.5)
     (self-alignment-X . 0)
     (self-alignment-Y . 0)
     (script-priority . 100)
```
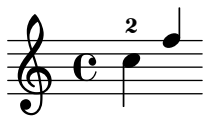
```
(stencil . ,ly:text-interface::print)
(direction . ,ly:script-interface::calc-direction)
(font-encoding . fetaNumber)
(font-size . -5)  ; don't overlap when next to heads.
(meta . ((class . Item)
(interfaces . (finger-interface
               font-interface
               text-script-interface
               text-interface
               side-position-interface
               self-alignment-interface
               item-interface))))))
```

Como podemos ver, el objeto `Fingering` no es más que un montón de valores de variables, y la página web de la Referencia de funcionamiento interno se genera directamente a partir de esta definición.

5.2.3 Determinar la propiedad del grob

Recordemos que queríamos cambiar la posición del **2** en

```
c-2
\stemUp
f
```



Puesto que el **2** se encuentra colocado verticalmente sobre su nota, tenemos que negociar con el interfaz asociado con esta colocación. Esto se hace usando `side-position-interface`. La página que describe este interface dice:

`side-position-interface`

Colocar un objeto víctima (este mismo) junto a otros objetos (el soporte). La propiedad `direction` significa dónde poner el objeto víctima con relación al soporte (¿a la izquierda o a la derecha, encima o debajo?)

Debajo de esta descripción, la variable `padding` (relleno) se describe como

`padding` (dimensión, en espacios de pentagrama)
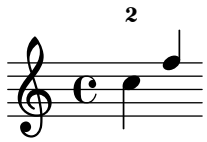
Añadir esta cantidad de espacio adicional entre objetos que están unos junto a otros.

Aumentando el valor de `padding`, podemos alejar la cifra de digitación de la cabeza de la nota. La siguiente orden inserta un espacio en blanco de 3 espacios de pentagrama entre la nota y la digitación:

```
\once \override Voice.Fingering #'padding = #3
```

Al insertar esta instrucción antes de que se haya creado el objeto Fingering, es decir, antes del `c2`, llegamos al siguiente resultado:

```
\once \override Voice.Fingering #'padding = #3
c-2
\stemUp
f
```



En este caso, el contexto de este truco es `Voice`. Este hecho se puede deducir también a partir de la referencia del programa, ya que la página dedicada al añadido Sección “Fingering_engraver” in *Referencia de Funcionamiento Interno* dice

El grabador Fingering_engraver es parte de los contextos: . . . Sección “Voice” in *Referencia de Funcionamiento Interno*

5.2.4 Convenciones de nombres

5.3 Modificar las propiedades

5.3.1 Panorámica de la modificación de las propiedades

Cada contexto es responsable de la creación de ciertos tipos de objetos gráficos. Los ajustes que se usan para imprimir estos objetos también se almacenan por contexto. Mediante la modificación de estos ajustes, se puede alterar la apariencia de los objetos.

La sintaxis de esto es

```
\override contexto.nombre #'propiedad = #valor
```
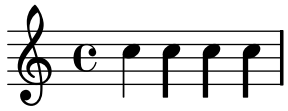
Aquí *nombre* es el nombre de un objeto gráfico, como `Stem` o `NoteHead`, y *propiedad* es una variable interna del sistema de formateo (‘propiedad del grob’ o ‘propiedad de disposición’). Este último es un símbolo, y por ello debe ir precedido de un apóstrofo. La subsección Sección 5.3 [Modificar las propiedades], página 400 explica cómo se deben cumplimentar los conceptos *nombre*, *propiedad* y *valor*. Aquí sólo nos ocuparemos de la funcionalidad des esta instrucción.

La instrucción

```
\override Staff.Stem #'thickness = #4.0
```

hace más gruesas las plicas (el valor predeterminado es 1.3, con el grosor de una línea del pentagrama como unidad). Puesto que la instrucción especifica como contexto a `Staff`, sólo se aplica al pentagrama actual. Otros pentagramas mantienen su aspecto normal. Aquí vemos la instrucción en pleno funcionamiento:
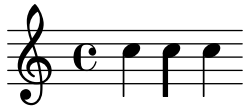
```
c4
\override Staff.Stem #'thickness = #4.0
c4
c4
c4
```



La instrucción `\override` modifica la definición de la plica `Stem` dentro del pentagrama en curso `Staff`. Después de que la instrucción se ha interpretado, todas las plicas se engrosan.

De manera análoga a `\set`, el argumento *contexto* se puede omitir, ocasionando que se utilice el contexto predeterminado `Voice`. Al añadir `\once` se aplica el cambio durante un solo paso de tiempo.

```
c4
\once \override Stem #'thickness = #4.0
c4
c4
```



El `\override` se debe hacer antes de que el objeto se inicia. Por tanto, al alterar objetos *Spanner* «de extensión» como ligaduras o barras, la instrucción `\override` se debe ejecutar en el momento en que se crea el objeto. En este ejemplo:

```
\override Slur #'thickness = #3.0
c8[( c
\override Beam #'thickness = #0.6
c8 c])
```



la ligadura es más gruesa pero la barra no lo es. Esto es así porque la instrucción para `Beam`, la barra, va después de que la barra se ha iniciado, y por ello no tiene ningún efecto.

De forma análoga a `\unset`, la instrucción `\revert` para un contexto deshace una instrucción `\override`; como con `\unset`, solamente afecta a los ajustes que se hicieron dentro del mismo contexto. En otras palabras, el `\revert` del siguiente ejemplo no hace nada.

```
\override Voice.Stem #'thickness = #4.0
\revert Staff.Stem #'thickness
```

Algunas opciones «trucables» se llaman 'subpropiedades' y residen dentro de las propiedades normales. Para trucarlas, utilice instrucciones de la forma

```
\override context.name #'property #'subproperty = #value
```

tales como

```
\override Stem #'(details beamed-lengths) = #'(4 4 3)
```

Véase también

Referencia de funcionamiento interno: Sección "OverrideProperty" in *Referencia de Funcionamiento Interno*, Sección "RevertProperty" in *Referencia de Funcionamiento Interno*, Sección "PropertySet" in *Referencia de Funcionamiento Interno*, Sección "Backend" in *Referencia de Funcionamiento Interno*, Sección "All layout objects" in *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

El «back-end» o motor de salida no es muy estricto en la comprobación de tipos de las propiedades de objetos. Las referencias cíclicas en valores Scheme de propiedades pueden producir cuelgues o salidas abruptas, o las dos cosas.
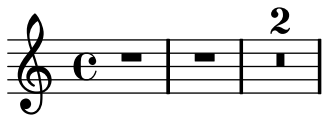
5.3.2 La instrucción `\set`

Cada contexto puede tener distintas *propiedades*, variables contenidas dentro de ese contexto. Se pueden cambiar mientras dura el paso de interpretación. Se consigue insertando la instrucción `\set` dentro de la música:

```
\set contexto.propiedad = #valor
```

Por ejemplo:
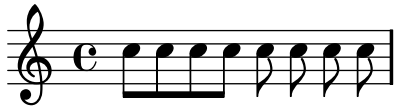
```
R1*2
\set Score.skipBars = ##t
R1*2
```



Estas instrucción salta los compases que no tienen notas. El resultado es que los silencios multicompás se condensan. El valor asignado es un objeto de Scheme. En este caso, es `#t`, el valor booleano True o verdadero.
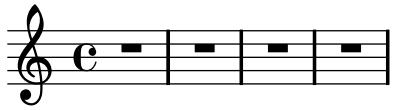
Si se omite el argumento *context*, entonces se utiliza el contexto actual de nivel más bajo (normalmente `ChordNames`, `Voice` o `Lyrics`). En este ejemplo:

```
c8 c c c
\set autoBeaming = ##f
c8 c c c
```



el argumento *contexto* de la instrucción `\set` se omite, así pues el barrado automático se desactiva en la voz actual. Observe que el contexto de nivel más bajo no siempre contiene la propiedad que queremos cambiar (por ejemplo, no tendrá ningún efecto intentar establecer un valor para la propiedad `skipBars` del contexto de nivel más bajo, en este caso `Voice`).

```
R1*2
\set skipBars = ##t
R1*2
```



Los contextos son jerárquicos, y si se ha especificado un contexto mayor, por ejemplo `Staff`, entonces el cambio se aplicaría también a todos los contextos `Voice` dentro del pentagrama actual. El cambio se aplica 'al vuelo', durante la música, de manera que el ajuste sólo afecta al segundo grupo de corcheas.

También existe una instrucción `\unset`:

`\unset` *`contexto.propiedad`*

que elimina la definición de *propiedad*. Esta instrucción elimina la definición solamente si está establecida dentro de *contexto*, de manera que

```
\set Staff.autoBeaming = ##f
```

introduce un ajuste de la propiedad en el nivel de `Staff`. El ajuste también se aplica a la `Voice` actual. Sin embargo:

```
\unset Voice.autoBeaming
```

no tiene ningúun efecto. Para cancelar este ajuste, se debe especificar el `\unset` en el mismo nivel que el `\set` original. Dicho de otra forma, para deshacer el efecto de `Staff.autoBeaming = ##f` se necesita

```
\unset Staff.autoBeaming
```
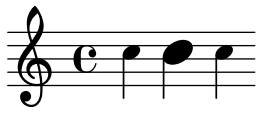
Como `\set`, el argumento *contexto* no se tiene que especificar para un contexto del nivel más bajo, por lo que los dos enunciados

```
\set Voice.autoBeaming = ##t
\set autoBeaming = ##t
```

son equivalentes.

Los ajustes que se aplican solamente a un único paso de tiempo se pueden escribir con `\once`, por ejemplo en

```
c4
\once \set fontSize = #4.7
c4
c4
```



la propiedad `fontSize` se deshace automáticamente después de la segunda nota.

En el manual de Referencia de funcionamiento interno hay una descripción completa de todas las propiedades de contexto disponibles, consulte Traducción ↦ Propiedades de contexto modificables por el usuario.

5.3.3 La instrucción `\override`

Las instrucciones que modifican la salida tienen por lo general un aspecto como

```
\override Voice.Stem #'thickness = #3.0
```

Para construir este truco debemos determinar los siguientes datos:

- el contexto: aquí `Voice`.
- el objeto de presentación: aquí `Stem`.
- la propiedad de presentación: aquí `thickness`.
- un valor adecuado: aquí `3.0`.

Ciertas opciones ajustables se denominan 'subpropiedades' y residen en el interior de las propiedades normales. Para modificarlas utilice instrucciones de la forma

```
\override Stem #'(details beamed-lengths) = #'(4 4 3)
```

Para muchas propiedades, independientemente del tipo de datos de la propiedad, si se establece el valor de la propiedad a falso (`##f`) se producirá su desactivación, ocasionando que LilyPond ignore por completo dicha propiedad. Esto es especialmente útil para «apagar» propiedades de grobs (objetos gráficos) que de otra manera causarína problemas.
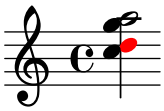
Mostraremos a continuación cómo localizar esta información en el manual de notación y en la referencia de funcionamiento interno.

5.3.4 La instrucción `\tweak`

En ocasiones es posible tomar un atajo para realizar el ajuste fino de los objetos gráficos. Para objetos que reusultan directamente de un elemento de código de la entrada, puede usar la función `\tweak`, por ejemplo

```
< c
  \tweak #'color #red
  d
  g
  \tweak #'duration-log #1
  a
> 4
-\tweak #'padding #8
-^
```
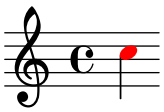
^



Pero el uso principal de la instrucción `\tweak` es modificar solamente uno de varios elementos de notación que dan comienzo en el mismo momento musical, como las notas de un acorde, o corchetes de tresillo que empiezan al mismo tiempo.

Para ver una introducción a la sintaxis y los usos de la instrucción tweak, consulte Sección "Tweaking methods" in *Manual de Aprendizaje*.

La instrucción `\tweak` fija una propiedad en el objeto que viene a continuación de forma directa, sin necesidad de especificar el nombre del grob o el contexto. Para que esto funcione, es necesario que la instrucción `\tweak` permanezca inmediatamente adyacente al objeto al que se aplica, después de que el archivo de entrada se ha convertido en un flujo musical. Con frecuencia no es el caso, pues muchos elementos adicionales se insertan en la corriente musical de forma implícita. Por ejemplo, cuando se procesa una nota que no forma parte de un acorde, LilyPond inserta implícitamente un evento `ChordEvent` antes de la nota, separando así el truco de la nota. Sin embargo, si los símbolos de acorde se sitúan cerca del truco y la nota, la instrucción `\tweak` viene después del `ChordEvent` en el flujo musical, permaneciendo así adyacente a la nota, y con la posibilidad de modificarla.

Así, esto funciona:

```
<\tweak #'color #red c>4
```



pero esto no funciona:

```
\tweak #'color #red c4
```



Si se colocan varios elementos similares en el mismo momento musical, la instrucción `\override` no se puede usar para modificar uno solo de ellos: aquí es donde se debe usar la instrucción `\tweak`. Entre los elementos que pueden aparecer más de una vez en el mismo momento musical están los siguientes:

- las cabezas de las notas de un acorde
- signos de articulación sobre la misma nota
- ligaduras de unión entre notas de un acorde
- corchetes de grupos especials que comienzan en el mismo momento

y se puede usar `\tweak` para modificar cualquier aparición específica de estos elementos.

Es de resaltar que la instrucción `\tweak` no se puede usar para modificar plicas, barras o alteraciones accidentales, porque éstos se generan más tarde por parte de las cabezas de nota, en vez de por elementos musicales que están en el flujo de entrada. Ni se puede usar una instrucción `\tweak` para modificar claves o indicaciones de compás, ya que éstos están separados de cualquier instrucción `\tweak` precedente en el flujo de entrada a causa de la inserción automática de elementos adicionales necesarios para especificar el contexto.

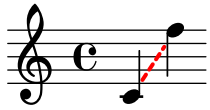
Pero la instrucción `\tweak` se puede usar como alternativa a la instrucción `\override` para modificar estos elementos notacionales que no producen la adición de ningún elemento notacional implícito antes de ellos mismos en el flujo musical. Por ejemplo, las ligaduras de expresión se pueden modificar de esta forma:

```
c-\tweak #'thickness #5 ( d e f)
```



También se pueden escribir varias instrucciones `\tweak` antes de un elemento notacional, y todas ellas le afectan:

```
c
-\tweak #'style #'dashed-line
-\tweak #'dash-fraction #0.2
-\tweak #'thickness #3
-\tweak #'color #red
 \glissando
f'
```



El flujo musical que se genera a partir de una sección de un archivo de entrada, incluido cualquier elemento insertado automáticamente, puede examinarse, véase Sección 6.3.1 [Presentación de expresiones musicales], página 438. Esto puede ser de utilidad en la determinación de lo que puede modificarse por medio de una instrucción `\tweak`.

Véase también

Manual de aprendizaje: Sección “Tweaking methods” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 6.3.1 [Presentación de expresiones musicales], página 438.

Advertencias y problemas conocidos

La instrucción `\tweak` no se puede usar dentro de una variable.

Las instrucciones `\tweak` no se pueden usar dentro del modo `\lyricmode`.

La instrucción `\tweak` no se puede usar para modificar los puntos de control de una única ligadura entre varias dentro de un acorde, aparte de la primera que aparece en el archivo de entrada.

5.3.5 `\set` frente a `\override`

Hemos visto dos formas de cambiar las propiedades: `\set` y `\override`. De hecho, en realidad existen dos clases diferentes de propiedades.

Los contextos pueden tener propiedades, que por lo general reciben nombres en `mayúsculasDeCamello`. Principalmente controlan la traducción de la música a la notación, p.ej. `localKeySignature` (para determinar si hay que imprimir las alteraciones accidentales), `measurePosition` (para determinar cuándo imprimir una línea divisoria). Las propiedades de contexto pueden ver modificado su valor con el tiempo según se interpreta una pieza de música; `measurePosition` es un ejemplo obvio de esto. Las propiedades de contexto se modifican con `\set`.

Hay un tipo especial de propiedad de contexto: la descripción del elemento. Estas propiedades reciben nombres en `MayúsculasDeCamello` (comenzando en letra mayúscula). Contienen los 'ajustes por defecto' para dicho elemento gráfico como una lista asociativa. Consulte '`scm/define-grobs.scm`' para ver qué tipos de ajustes hay. Las descripciones de los elementos se pueden modificar con `\override`.

Realmente, `\override` es un atajo;

`\override` *`contexto.nombre #'propiedad = #valor`*

es más o menos equivalente a

`\set` *`contexto.nombre #'propiedad = #(cons (cons 'propiedad valor) <valor previo de contexto)`*

El valor de `contexto` (la lista-a) se usa para da un valor inicial a las propiedades de los objetos gráficos individuales. Los objetos gráficos también tienen propiedades, que reciben nombres en el estilo de Scheme, con `palabras-con-guiones`. Los valores de las propiedades de objetos gráficos cambian durante el proceso de formateo: el formateo básicamente consiste en calcular las propiedades utilizando funciones de callback.

`fontSize` es una propiedad especial: equivale a escribir `\override ... #'font-size` para todos los objetos pertinentes. Al ser éste un cambio muy común, se creó la propiedad especial (modificada con `\set`).

5.4 Conceptos y propiedades útiles

5.4.1 Modos de entrada

La forma en que se interpreta la notación conenida dentro de un archivo de entrada, está determinada por el modo de entrada en curso.

Modo de acordes

Se activa con la instrucción `\chordmode` y produce que la entrada se interprete con al sintaxis de la notación de acordes, véase Sección 2.7 [Notación de acordes], página 264. Los acordes se imprimen como notas sobre un pentagrama.

El modo de acordes se activa también con la instrucción `\chords`. Esto crea también un contexto `ChordNames` nuevo y produce que el código que sigue se interprete conla sintaxis de la notación de acordes y se imprima como nombres de acorde dentro del contexto `ChordNames`, véase [Impresión de los nombres de acorde], página 269.

Modo de percusión

Se activa con la instrucción `\drummode` y produce que el código de entrada se interprete con la sintaxis de la notación de percusión, véase [Notación básica de percusión], página 251.

El modo de percusión también se activa con la instrucción `\drums`. También crea un contexto `DrumStaff` nuevo y hace que el código que sigue se interprete con la sintaxis de la notación de percusión y se imprima como símbolos de percusión sobre un pentagrama de percusión, véase [Notación básica de percusión], página 251.

Modo de cifras

Se activa con la instrucción `\figuremode` y hace que el código de entrada se interprete con la sintaxis del bajo cifrado, véase [Introducir el bajo cifrado], página 277.

El modo de cifrase también se activa con la instrucción `\figures`. También crea un contexto de `FiguredBass` nuevo y hace que el código que viene a continuación se interprete con la sintaxis del bajo cifrado y se imprima como símbolos de bajo cifrado dentro del contexto `FiguredBass`, véase [Introducción al bajo cifrado], página 276.

Modos de traste y tablatura

No existen modos de entrada especiales para introducir símibolos de trastes y de tablatura.

Para crear diagramas de trastes, escriba las notas o acordes en el modo de notas e imprímalos dentro de un contexto `TabStaff`, véase [Tablaturas predeterminadas], página 225.

Para crear diagramas de trastes encima de un pentagrama, escríbalos como elementos de marcado encima de las notas utilizando la instrucción `\fret-diagram`, véase [Marcas de diagramas de trastes], página 229.

Modo de letra

Se activa con la instrucción `\lyricmode`, y hace que la entrada se interprete como sílabas de la letra de la canción con duraciones opcionales y modificadores de letra asociados, véase Sección 2.1 [Música vocal], página 190.

El modo de letra también se habilita con la instrucción `\addlyrics`. Esto también crea un contexto `Lyrics` nuevo y una instrucción `\lyricsto` implícita que asocia la letra que viene a continuación con la música precedente.

Modo de marcado

Se activa con la instrucción `\markup`, y hace que la entrada se interprete con la sintaxis del marcado, véase Sección B.8 [Text markup commands], página 476.

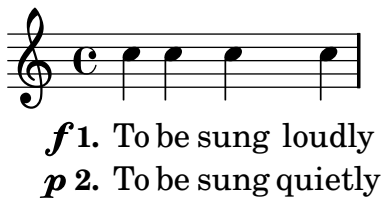
Modo de notas

Es el modo predeterminado o se puede activar con la instrucción `\notemode`. La entrada se interpreta como alturas, duraciones, marcado, etc. y se imprime como notación musical sobre un pentagrama.

Normalmente no es necesario especificar el modo de notas de forma explícita, pero puede ser útil hacerlo en ciertas situaciones, por ejemplo si estamos en el modo de letra, en el modo de acordes o en otro modo y queremos insertar algo que solamente se puede hacer con la sintaxis del modo de notas.

Por ejemplo, para insertar indicaciones dinámicas para las estrofas de una pieza coral es necesario entrar en el modo de notas para poder interpretar dichas indicaciones:

```
{ c4 c4 c4 c4 }
\addlyrics {
  \notemode{\set stanza = \markup{ \dynamic f 1. } }
  To be sung loudly
}
\addlyrics {
  \notemode{\set stanza = \markup{ \dynamic p 2. } }
  To be sung quietly
}
```



5.4.2 Dirección y posición

Al tipografiar música, la dirección y colocación de muchos elementos es cuestión de elección. Por ejemplo, las plicas de las notas se pueden dirigir hacia arriba o hacia abajo; la letra, las indicaciones dinámicas y otras marcas expresivas se pueden colocar encima o debajo del pentagrama; el texto se pude alinear a la izquierda, a la derecha o centrado; etc. La mayoría de estas elecciones pueden dejarse que LilyPond las determine automáticamente, pero en ciertos casos puede ser deseable forzar una dirección o colocación concreta.

Acciones predeterminadas

De forma predeterminada algunas direcciones siempre son hacia arriba o siempre hacia abajo (p. ej. los matices o el calderón), mientras que otras cosas pueden alternar entre arriba y abajo en función de la dirección de las plicas (como las ligaduras o los acentos).

Disposición de contexto

Los contextos se colocan dentro de un sistema de arriba a abajo en el orden en que se encuentran. Sin embargo, observe que se crea un contexto implícitamente si se encuentra una instrucción cuando no está disponible un contexto apropiado para contenerla.

Se pude cambiar el orden predeterminado en que los contextos se presentan, véase Sección 5.1.6 [Alinear los contextos], página 395

Indicadores de dirección de las articulaciones

Al añadir articulaciones a notas se puede omitir normalmente el indicador de dirección, `^` (que significa "arriba"), `_` (que significa "abajo") o `-` (que significa "usar la dirección predeterminada"), en cuyo caso se supone el perdeterminado `-`. Pero se necesita un indicador de dirección **always** antes de:

- las instrucciones `\tweak`
- las instrucciones `\markup`
- las instrucciones `\tag`
- los marcados de cadena, p.ej. `-"cadena"`
- las instrucciones de digitación, p.ej. `-1`
- las abreviaturas de articulación, p.ej. `-.`, `->`, `--`

La propiedad de dirección

La posición o dirección de muchos objetos de presentación está controlada por la propiedad `direction`.

El valor de la propiedad `direction` se puede establecer al valor `1`, con el significado de "hacia arriba" o "encima", o a `-1`, con el significado de "hacia abajo" o "debajo". Se pueden usar los símbolos `UP` y `DOWN` en sustitución de `1` y `-1` respectivamente. La dirección predeterminada se puede especificar estableciendo `direction` a `0` ó a `CENTER`. De forma alternativa, en muchos casos existen instrucciones predefinidas para especificar la dirección. Todas ellas son de la forma:

`\xxxUp`, `xxxDown`, `xxxNeutral`

donde `xxxNeutral` significa "utilizar la dirección predeterminada". Véase Sección "Within-staff objects" in *Manual de Aprendizaje*.

En alguna que otra ocasión, con el único ejemplo usual del arpegio, el valor de la propiedad `direction` especifica si el objeto se debe colocar a la izquierda o a la derecha del objeto padre. En este caso `-1` ó `LEFT` significan "a la izquierda" y `1` ó `RIGHT` significan "a la derecha". `0` ó `CENTER` significan "utilizar la dirección predeterminada", como antes.

5.4.3 Distancias y medidas

Las distancias en LilyPond son de dos tipos: absolutas y escaladas.

Las distancias absolutas se usan para especificar márgenes, sangrados y otros detalles de diseño de página, y de forma predeterminada se especifican en milímetros. Las distancias se pueden especificar en otras unidades escribiendo después de la cifra indicativa de la cantidad, `\mm`, `\cm`, `\in` (pulgadas), o `\pt` (puntos, 1/72.27 pulgadas). Las distancias de diseño de página se pueden especificar también en unidades escalables (véase el párrafo siguiente) adjuntando `\staff-space` a la cantidad. La disposición de página se describe en detalle en Sección 4.1.2 [Formateo de las páginas], página 346.

Las distancias escaladas siempre se especifican en unidades de un espacio del pentagrama o, más raramente, medio espacio del pentagrama. El espacio de pentagrama es la distancia entre dos líneas del pentagrama adyacentes. El valor predeterminado se puede cambiar globalmente fijando el tamaño global del pentagrama, o se puede sobreescribir localmente cambiando la propiedad `staff-space` del objeto `StaffSymbol`. Las distancias escaladas se escalan automáticamente con cualquier cambio al tamaño global del pentagrama o a la propiedad `staff-space` del objeto `StaffSymbol`, pero las fuentes tipográficas se escalan solamente con los cambios efectuados al tamaño global del pentagrama. Así, el tamaño global del pentagrama posibilita la fácil variación del tamaño general de una partitura impresa. Para ver los métodos de establecimiento del tamaño global del pentagrama, véase Sección 4.2.1 [Establecer el tamaño del pentagrama], página 351.

Si se necesita dibujar sólo una sección de una partitura a una escala distinta, por ejemplo una sección ossia o una nota al pie, no se puede simplemente cambiar el tamaño global del pentagrama porque esto afectaría a toda la partitura. En tales casos, el cambio de tamaño se hace sobreescribiendo tanto la propiedad `staff-space` de `StaffSymbol` como el tamaño de las fuentes tipográficas. Está a nuestra disposición una función de Scheme, `magstep`, para convertir de un cambio en el tamaño de la fuente al cambio equivalente en `staff-space`. Para ver una explicación y un ejemplo de su utilización, consulte Sección “Length and thickness of objects” in *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección “Length and thickness of objects” in *Manual de Aprendizaje*.
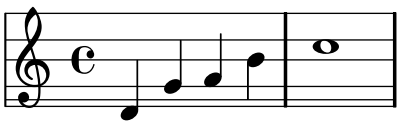
Referencia de la notación: Sección 4.1.2 [Formateo de las páginas], página 346, Sección 4.2.1 [Establecer el tamaño del pentagrama], página 351.

5.4.4 Propiedades del símbolo del pentagrama

Se puede definir al mismo tiempo la posición vertical de las líneas de la pauta y el número de líneas de la misma. Como muestra el siguiente ejemplo, las posiciones de las notas no están influidas por las posiciones de las líneas de la pauta.
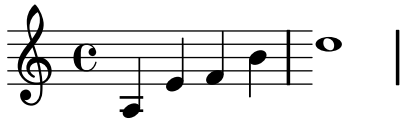
Nota: La propiedad `'line-positions` sobreescribe a la propiedad `'line-count`. El número de líneas de la pauta está definido implícitamente por el número de elementos de la lista de valores de `'line-positions`.

```
\new Staff \with {
  \override StaffSymbol #'line-positions = #'(7 3 0 -4 -6 -7)
}
{ a4 e' f b | d1 }
```



Se puede modificar la anchura de la pauta. Las unidades son espacios de pentagrama. El espaciado de los objetos dentro del pentagrama no resulta afectado por este ajuste.

```
\new Staff \with {
  \override StaffSymbol #'width = #23
}
{ a4 e' f b | d1 }
```



5.4.5 Objetos de extensión

Muchos objetos de notación musical abarcan varias notas o incluso varios compases. Son ejemplos los crescendi, trinos, corchetes de grupo especial y corchetes de primera y segunda vez. Estos objetos se llaman "spanners" u «objetos de extensión», y tienen propiedades especiales para controlar su apariencia y comportamiento. Algunas de estas propiedades son comunes a todos los objetos de extensión; otras se limitan a un subconjunto de los extensores.

Todos los objetos de extensión contemplan el interface `spanner-interface`. Algunos, básicamente aquellos que trazan una línea recta entre los dos objetos, contemplan también el interface `line-spanner-interface`.

Uso del `spanner-interface`

Este interface proporciona dos propiedades que se aplican a varios extensores.
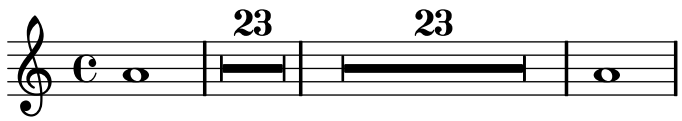
La propiedad `minimum-length`

La longitud mínima del objeto de extensión se pesoecifica a través de la propiedad `minimum-length`. Su aumento suele producir el efecto necesario de aumentar el espaciado de las notas entre los dos puntos extremos. Sin embargo, esta sobreescritura no tiene ningún efecto sobre muchos extensores, pues su longitud está determinada por otras consideraciones. Más abajo se muestran algunos ejemplos de dónde es efectiva.
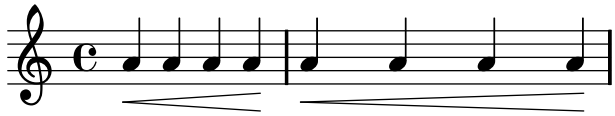
```
a~a
a
% aumentar la longitud de la ligadura
-\tweak #'minimum-length #5
~a
```



```
a1
\compressFullBarRests
R1*23
% aumentar la longitud del compás en silencio
\once \override MultiMeasureRest #'minimum-length = #20
R1*23
a1
```




```
a \< a a a \!
% aumentar la longitud del regulador
\override Hairpin #'minimum-length = #20
a \< a a a \!
```



Esta sobreescritura se puede usar también para aumentar la longitud de las ligaduras de expresión y de fraseo:

```
a( a)
a
-\tweak #'minimum-length #5
( a)

a\( a\)
a
-\tweak #'minimum-length #5
\( a\)
```
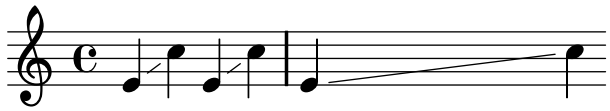


Para algunos objetos de preesentación, la propiedad `minimum-length` es efectiva sólo si se llama explícitamente al procedimiento `set-spacing-rods`. Para hacerlo, se debe fijar la propiedad `springs-and-rods` al valor `ly:spanner::set-spacing-rods`. Por ejemplo, la longitud mínima de un glissando no tiene efecto a no ser que se establezca la propiedad `springs-and-rods`:

```
% predeterminado
e \glissando c'

% no es efectivo por sí solo
\once \override Glissando #'minimum-length = #20
e, \glissando c'
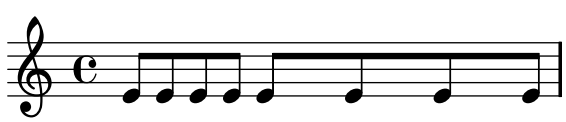
% es efectivo sólo cuando las dos sobreescrituras están presentes
\once \override Glissando #'minimum-length = #20
\once \override Glissando #'springs-and-rods = #ly:spanner::set-spacing-rods
e, \glissando c'
```



Lo mismo se puede decir del objeto `Beam`:

```
% no es efectivo por sí solo
\once \override Beam #'minimum-length = #20
e8 e e e

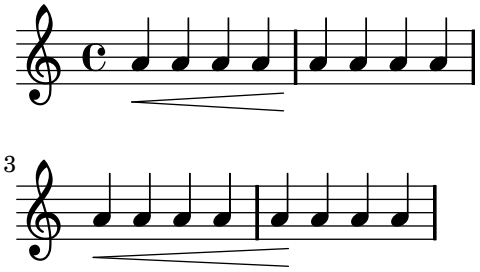
% es efectivo sólo cuando las dos sobreescrituras están presentes
\once \override Beam #'minimum-length = #20
\once \override Beam #'springs-and-rods = #ly:spanner::set-spacing-rods
e8 e e e
```



The `to-barline` *property*

La segunda propiedad útil del `spanner-interface` es `to-barline`. De forma predeterminada tiene el valor cierto, haciendo que los reguladores y otros objetos de extensión que terminan sobre la primera nota de un compás, en vez de eso terminen en la línea divisoria inmediatamente precedente. Si se establece al valor falso, el extensor llegará más allá de la barra de compás y terminará excatamente sobre la nota:

```
a \< a a a a \! a a a \break
\override Hairpin #'to-barline = ##f
a \< a a a a \! a a a
```



Esta propiedad no es efectiva para todos los extensores. Por ejemplo, su establecimiento a `#t` no tienen ningún efecto sobre las ligaduras de expresión o de fraseo, o sobre otros extensores para los que terminar en la barra de compás no tendría ningún significado.
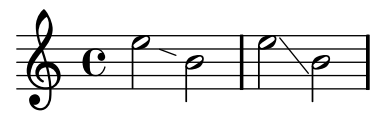
Uso del `line-spanner-interface`

Entre los objetos que contemplan el interface `line-spanner-interface` se encuentran

- `DynamicTextSpanner`
- `Glissando`
- `TextSpanner`
- `TrillSpanner`
- `VoiceFollower`

La rutina responsable de dibujar los sellos de estos extensores es `ly:line-interface::print`. esta rutina determina la localización exacta de los dos puntos extremos y traza una línea entre ellos, en el estilo solicitado. Las posiciones de los dos puntos extremos del extensor se calculan al vuelo, pero es posible sobreescribir sus coordenadas Y. Las propiedades que se deben especificar están anidadas a dos niveles de profundidad en la jerarquía de propiedades, pero la sintaxis de la instrucción `\override` es bastante sencilla:

```
e2 \glissando b
\once \override Glissando #'(bound-details left Y) = #3
\once \override Glissando #'(bound-details right Y) = #-2
e2 \glissando b
```

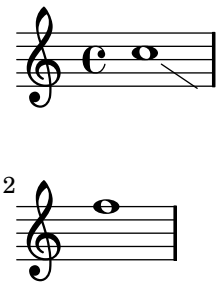


Las unidades para la propiedad `Y` son `staff-space`s, siendo el punto del cero la línea central del pentagrama. Para el glissando, esto es el valor de `Y` en la coordenada X que corresponde al punto central de cada cabeza de nota si nos imaginamos que la línea se extiende hasta allí.

Si no está fijado `Y`, su valor se calcula a partir de la posición vertical del punto de anclaje correspondiente del extensor.

En caso de salto de línea, los valores para los puntos extremos se especifican por las sub-listas `left-broken` y `right-broken` de `bound-details`. Por ejemplo:

```
\override Glissando #'breakable = ##t
\override Glissando #'(bound-details right-broken Y) = #-3
c1 \glissando \break
f1
```



Un número de propiedades adicionales de las sub-listas `left` y `right` de la propiedad `bound-details` se pueden especificar de la misma forma que `Y`:

`Y` Establece la coordenada Y del punto extremo, en desplazamientos de `staff-space`s desde la línea central del pentagrama. De forma predeterminada es el centro del objeto ancla, y así un glissando apunta al centro vertical de la cabeza de la nota.

Para extensores horizontales como los extensores de texto y los trinos, está inamoviblemente codificado como 0.
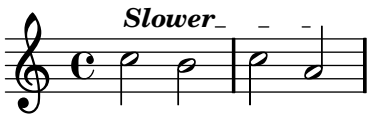
`attach-dir (dirección de anclaje)`
Determina dónde comienza y termina la línea en la dirección X, con relación al objeto ancla. Sí, un valor de `-1` (o `LEFT`, izquierda) hace que la línea comienze o termine en el lado izquierdo de la cabeza de la nota a la que está anclado.

`X` Es la coordenada X absoluta del punto extremo. Se suele calcular al vuelo, y su sobreescritura no tiene un efecto útil.

`stencil (sello)`
Los extensores de línea pueden tener símbolos al comienzo o al final, lo que está contenido en esta sub-propiedad. Esto es para uso interno; se recomienda en su lugar el uso de `text`.

`text (texto)`
Es un elemento de marcado que se evalúa para dar lugar al sello. Se usa para escribir *cresc.*, *tr* y otros textos sobre los objetos de extensión horizontales.
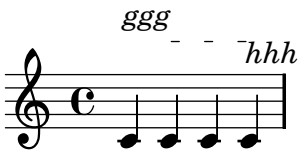
```
\override TextSpanner #'(bound-details left text)
   = \markup { \small \bold Slower }
c2\startTextSpan b c a\stopTextSpan
```



`stencil-align-dir-y (alineación del sello en y)`
`stencil-offset (desplazamiento del sello)`
Si no se modifican uno u otro, el sello se coloca sencillamente en el punto extremo, centrado sobrer la línea, como viene definido por las subpropiedades `X` e `Y`. Si se fijan `stencil-align-dir-y` o `stencil-offset` se moverá el símbolo del borde verticalmente con relación al extremo de la línea:

```
\override TextSpanner
  #'(bound-details left stencil-align-dir-y) = #-2
\override TextSpanner
  #'(bound-details right stencil-align-dir-y) = #UP

\override TextSpanner
  #'(bound-details left text) = #"ggg"
\override TextSpanner
  #'(bound-details right text) = #"hhh"
c4^\startTextSpan c c c \stopTextSpan
```
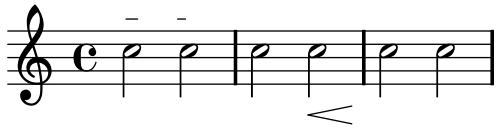


Observe que los valores negativos mueven el texto *hacia arriba*, al contrario de lo que podríoa esperarse, pues el valor de `-1` o `DOWN` (abajo) significa alinear el bborde *inferior* del texto con la línea de extensión. Un valor de `1` o `UP` (arriba) alinea el borde superior del texto con la línea extensora.

`arrow (flecha)`
: Al establecer esta sub-propiedad a `#t` se produce una punta de flecha en el extremo de la línea.

`padding (relleno)`
: Esta sub-propiedad controla el espacio entre el punto extremo especificado de la línea y el extremo real. Sin relleno, un glissando empezaría y terminaría en el centro de la cabeza de las notas.

La función musical `\endSpanners` finaliza de forma prematura el extensor que comienza sobre la nota que sigue inmediatamente a continuación. Se termina después de una nota exactamente, o en la siguiente barra de compás si `to-barline` es verdadero y se produce una divisoria antes de la siguiente nota.

```
\endSpanners
c2 \startTextSpan c2 c2
\endSpanners
c2 \< c2 c2
```



Si se usa `\endSpanners` no es necesario cerrar \startTextSpan con \stopTextSpan, ni cerrar los reguladores con `\!`.

Véase también

Referencia de funcionamiento interno: Sección “TextSpanner” in *Referencia de Funcionamiento Interno*, Sección “Glissando” in *Referencia de Funcionamiento Interno*, Sección “VoiceFollower” in *Referencia de Funcionamiento Interno*, Sección “TrillSpanner” in *Referencia de Funcionamiento Interno*, Sección “line-spanner-interface” in *Referencia de Funcionamiento Interno*.

5.4.6 Visibilidad de los objetos

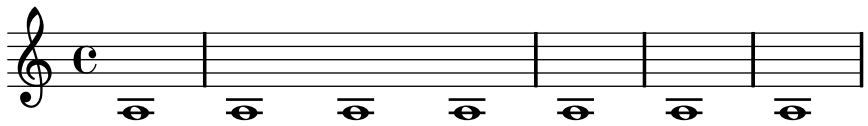
Hay cuatro formas principales en que se puede controlar la visibilidad de los objetos de presentación: se puede eliminar su sello, se pueden volver transparentes, se pueden pintar de blanco, o se puede sobreescribir su propiedad `break-visibility`. Las tres primeras se aplican a todos los objetos de presentación; la última sólo a unos pocos: los objetos *divisibles*. El Manual de aprendizaje introduce estas cuatro técnicas, véase Sección "Visibility and color of objects" in *Manual de Aprendizaje*.

Hay también algunas otras técnicas que son específicas de ciertos objetos de presentación. Se estudian bajo Consideraciones especiales.

Eliminar el sello

Todo objeto de presentación tiene una propiedad stencil (sello). De forma predeterminada está establecida a la función específica que dibuja ese objeto. Se se sobreescribe eesta propiedad a `#f` no se llama a ninguna función y el objeto no se dibuja. La acción predeterminada se puede recuperar con `\revert`.
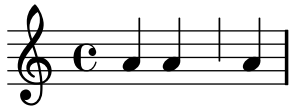
```
a1 a
\override Score.BarLine #'stencil = ##f
a a
\revert Score.BarLine #'stencil
a a a
```



Hacer transparentes los objetos

Todo objeto de presentación tiene una propiedad transparent (transparente) que de forma predeterminada está establecida a `#f`. Si se fija a `#t` el objeto aún ocupa espacio pero es invisible.
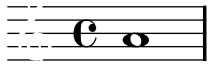
```
a4 a
\once \override NoteHead #'transparent = ##t
a a
```



Pintar los objetos de blanco

Todo objeto de presentación tiene una propiedad de color que de forma predeterminada está establecida a `black` (negro). Si se sobreescribe a `white` (blanco) el objeto será indistinguible del fondo blanco. Sin embargo, si el objeto cruza a otros objetos, el color de los puntos de cruce queda determinado por el orden en que se dibujan estos objetos, lo que puede dejar una imagen fantasma del objeto blanco, como puede verse aquí:

```
\override Staff.Clef #'color = #white
a1
```



Se puede evitar esto cambiando el orden de impresión de los objetos. Todos los objetos de presentación tienen una propiedad `layer` (capa) que se debe establecer a un valor entero. Los objetos con el valor de `layer` más bajo se dibujan primero, después se dibujan los objetos con valores progresivamente mayores, y así los objetos con valores más altos se dibujan encima de los que tienen valores más bajos. De forma predeterminada, casi todos los objetos tienen asignado un valor `layer` de `1`, aunque algunos objetos, entre ellos el pentagrama y las líneas divisorias, `StaffSymbol` y `BarLine`, tienen asignado un calor de `0`. El orden de impresión de los objetos con el mismo valor de `layer` es indeterminado.

En el ejemplo de arriba, la clave blanca, con un valor `layer` predeterminado de `1`, se dibuja después de las líneas del pentragrama (valor `layer` predeterminado de `0`), sobreimpresionándolas. Para cambiarlo, se debe dar al objeto `Clef` un valor de `layer` más bajo, digamos `-1`, para que se dibuje antes:

```
\override Staff.Clef #'color = #white
\override Staff.Clef #'layer = #-1
a1
```



Uso de break-visibility (visibilidad en el salto)

Casi todos los objetos de presentación se imprimen una sola vez, pero algunos como las líneas divisorias, claves, indicaciones de compás y armaduras de tonalidad, se pueden tener que imprimir dos veces cuando se produce un salto de línea : una vez al final de la línea y otra al comienzo de la siguiente. Estos objetos reciben el nombre de *divisibles*, y tienen una propiedad, `break-visibility` (visibilidad en el salto), para controlar su visibilidad en las tres posiciones en que pueden aparecer: al comienzo de una línea, dentro de la línea si se produce un cambio, y al final de la línea si el cambio se produce en ese lugar.

Por ejemplo, la indicación de compás se imprime de forma predeterminada al comienzo de la primera línea y en ningún otro lugar a no ser que cambie, en cuyo caso se imprime en el punto en que se produce el cambio. Si este cambio se produce al final de una línea, la nueva indicación de compás se imprime al principio de la línea siguiente y también al final de la línea anterior como indicación de precaución.

Este comportamiento se controla por medio de la propiedad `break-visibility`, que se explica en Sección “Visibility and color of objects” in *Manual de Aprendizaje*. Esta propiedad toma un vector de tres valores booleanos que, por orden, determinan si el objeto se imprime al final, dentro, o al principio de la línea. O, para ser más exactos: antes del salto de línea, si no hay salto, o después del salto.

Como alternativa se puede especificar cualquiera de las ocho combinaciones mediante funciones predefinidas cuya definición está en ‘`scm/output-lib.scm`’, donde las tres últimas columnas indican si los objetos de presentación serán visibles en las posiciones que se muestran en el encabezamiento de cada columna:

Forma de función	**Forma de vector**	**Antes del salto**	**Si no hay salto**	**Después del salto**
`all-visible`	`'#(#t #t #t)`	sí	sí	sí
`begin-of-line-visible`	`'#(#f #f #t)`	no	no	sí

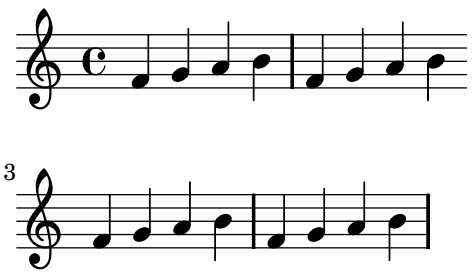
`center-visible`	`'#(#f #t #f)`	no	sí	no
`end-of-line-visible`	`'#(#t #f #f)`	sí	no	no
`begin-of-line-invisible`	`'#(#t #t #f)`	sí	sí	no
`center-invisible`	`'#(#t #f #t)`	sí	no	sí
`end-of-line-invisible`	`'#(#f #t #t)`	no	sí	sí
`all-invisible`	`'#(#f #f #f)`	no	no	no

Los ajustes predeterminados de `break-visibility` dependen del objeto de presentación. La tabla siguiente muestra todos los objetos de presentación de interés que resultan afectados por `break-visibility` y el ajuste predeterminado de esta propiedad:

Objeto	**Contexto usual**	**Valor predet.**
`BarLine`	`Score`	calculado
`BarNumber`	`Score`	`begin-of-line-visible`
`BreathingSign`	`Voice`	`begin-of-line-invisible`
`Clef`	`Staff`	`begin-of-line-visible`
`Custos`	`Staff`	`end-of-line-visible`
`DoublePercentRepeat`	`Voice`	`begin-of-line-invisible`
`KeySignature`	`Staff`	`begin-of-line-visible`
`OctavateEight`	`Staff`	`begin-of-line-visible`
`RehearsalMark`	`Score`	`end-of-line-invisible`
`TimeSignature`	`Staff`	`all-visible`

El ejemplo de abajo muestra el uso de la forma de vector para controlar la visibilidad de las líneas divisorias:

```
f4 g a b
f4 g a b
% Quitar la barra de compás al final de la línea en curso
\once \override Score.BarLine #'break-visibility = #'#(#f #t #t)
\break
f4 g a b
f4 g a b
```



Aunque los tres componentes del vector utilizado para sobreescribir `break-visibility` deben estar presentes, no todos son efectivos para todos los objetos de presentación, y algunas combinaciones pueden incluso dar errores. Son de aplicación las siguientes limitaciones:

- Las líneas divisorias no se pueden imprimir al principio de la línea.
- No se puede imprimir el número de compás al principio de la primera línea a no ser que su valor establecido sea distinto de 1.
- Clave: véase más abajo
- Las repeticiones dobles de tipo porcentaje se imprimen completamente o se suprimen por completo. Utilice begin-of line-invisible para imprimirlas y all-invisible para suprimirlas.

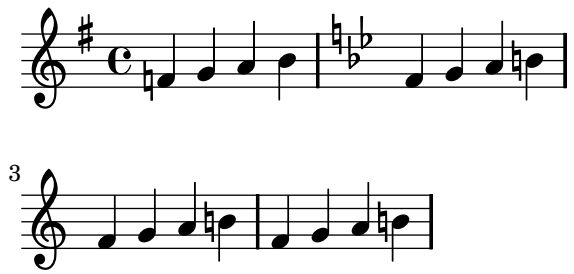
- Armadura: véase más abajo
- OctavateEight: véase más abajo

Consideraciones especiales

Visibilidad después de un cambio explícito
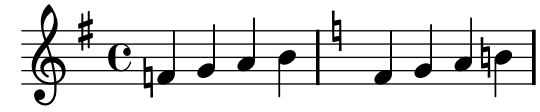
La propiedad `break-visibility` controla la visibilidad de las armaduras y cambios de clave sólo al principio de las líneas, es decir, después de un salto. No tiene ningún efecto sobre la visibilidad de la armadura o la clave después de un cambio explícito de tonalidad o de clave dentro o al final de una línea. En el ejemplo siguiente la armadura que sigue al cambio explícito de tonalidad a Si bemol mayor es visible incluso con `all-invisible` establecido.
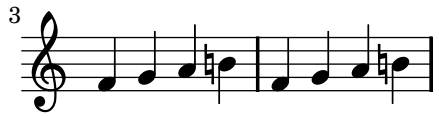
```
\key g \major
f4 g a b
% Intentar quitar todas las armaduras
\override Staff.KeySignature #'break-visibility = #all-invisible
\key bes \major
f4 g a b
\break
f4 g a b
f4 g a b
```



La visibilidad de estos cambios explícitos de tonalidad y de clave se controla por medio de las propiedades `explicitKeySignatureVisibility` y `explicitClefVisibility`. Son los equivalentes de la propiedad `break-visibility` y las dos toman un vector de tres valores booleanos o las funciones predefinidas relacionadas anteriormente, exactamente igual que `break-visibility`. Las dos son propiedades del contexto Staff, no de los propios objetos de presentación, y por tanto se establecen utilizando la instrucción `\set`. Las dos están establecidas de forma predeterminada al valor `all-visible`. Estas propiedades controlan sólo la visibilidad de las armaduras y las claves que resultan de los cambios explícitos y no afectan a las armaduras y tonalidades que están al principio de las líneas; para quitarlas, aún se debe sobreescribir la propiedad `break-visibility` en el objeto correspondiente.

```
\key g \major
f4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Staff.KeySignature #'break-visibility = #all-invisible
\key bes \major
f4 g a b \break
f4 g a b
f4 g a b
```
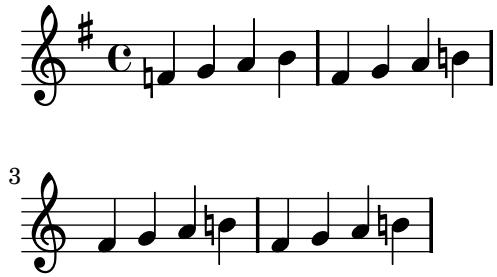




Visibilidad de las alteraciones de precaución

Para eliminar las alteraciones de precaución que se imprimen en un cambio de tonalidad explícito, establezca la propiedad `printKeyCancellation` del contexto Staff a `#f`:

```
\key g \major
f4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.printKeyCancellation = ##f
\override Staff.KeySignature #'break-visibility = #all-invisible
\key bes \major
f4 g a b \break
f4 g a b
f4 g a b
```



Con estas sobreescrituras solamente permanecen las alteraciones accidentales delante de las notas para indicar el cambio de tonalidad.

Automatic bars

Como caso especial, la impresión de las líneas divisorias también se puede inhabilitar estableciendo la propiedad `automaticBars` en el contexto Score. Si se fija a `#f`, las barras de compás no se imprimen automáticamente; se deben crear explícitamente con una instrucción `\bar`. A diferencia de la instrucción predefinida `\cadenzaOn`, los compases se siguen contando. La generación de compases continúa de acuerdo a esta cuenta si esta propiedad se establece posteriormente a `#t`. Si se fija al valor `#f`, sólo pueden producirse saltos de línea en instrucciones `\bar` explícitas.

Octavated clefs

El pequeño símbolo de octava sobre las claves en octava alta o baja se produce por parte del objeto `OctavateEight`. Su visibilidad se controla independientemente de la del objeto `Clef`, así que es necesario aplicar las sobreescrituras correspondientes `break-visibility` tanto a los objetos `Clef` como `OctavateEight` para suprimir completamente estos símbolos de clave al comienzo de cada línea.

Para los cambios de clave explícitos, la propiedad `explicitClefVisibility` controla tanto el símbolo de clave como el símbolo de octava asociado.

Véase también

Manual de aprendizaje: Sección “Visibility and color of objects” in *Manual de Aprendizaje*

5.4.7 Estilos de línea
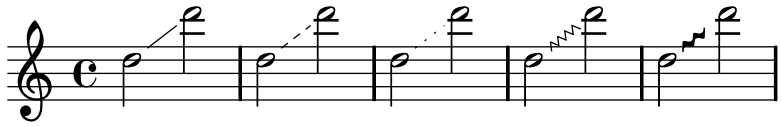
Ciertas indicaciones de ejecución, p.ej., *rallentando*, *accelerando* y los *trinos* se escriben como texto y se extienden sobre muchos compases mediante líneas, a veces punteadas u onduladas.

Todos ellos usan las mismas rutinas que el glissando para dibujar los textos y las líneas, y por ello el ajuste fino de su comportamiento se hace de la misma manera. Se hace con un

spanner (un objeto de extensión), y la rutina responsable de dibujar los objetos de extensión es `ly:line-interface::print`. Esta rutina determina la colocación exacta de los dos *extremos del objeto de extensión* y dibuja una línea entre ellos, en el estilo solicitado.
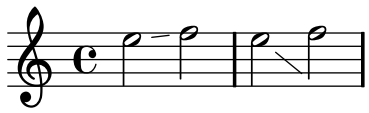
He aquí un ejemplo de los distintos estilos de línea disponibles, y cómo aplicarles ajustes finos.

```
d2 \glissando d'2
\once \override Glissando #'style = #'dashed-line
d,2 \glissando d'2
\override Glissando #'style = #'dotted-line
d,2 \glissando d'2
\override Glissando #'style = #'zigzag
d,2 \glissando d'2
\override Glissando #'style = #'trill
d,2 \glissando d'2
```



Las posiciones de los puntos extremos del objeto de extensión se computan al vuelo para cada uno de los objetos gráficos, pero es posible sobreescribirlos:

```
e2 \glissando f
\once \override Glissando #'(bound-details right Y) = #-2
e2 \glissando f
```



El valor de `Y` está establecido a `-2` para el extremo derecho. El lado izquierdo se puede ajustar de forma similar especificando `left` en vez de `right`.

Si no está establecido `Y`, el valor se computa a partir de la posición vertical de los puntos de anclaje izquierdo y derecho del extensor.

Son posibles otros ajustes de los extensores, para ver más detalles consulte Sección 5.4.5 [Objetos de extensión], página 410.

5.4.8 Rotación de objetos

Tanto los objetos de presentación como los elementos de texto de marcado se pueden girar cualquier ángulo respecto a cualquier punto, pero difiere el método de hacerlo.
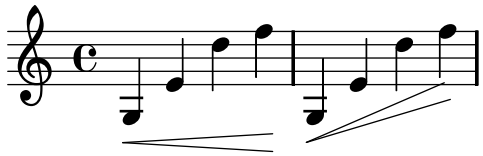
Rotación de objetos de presentación

Todos los objetos de presentación que contemplan el interface `grob-interface` se pueden rotar estableciendo su propiedad `rotation`. Acepta una lista de tres elementos: el ángulo de la rotación en sentido antihorario, y las coordenadas x e y del punto con relación al punto de referencia del objeto, alrededor del que se va a realizar la rotación. El ángulo de rotación se pespecifica en grados y las coordenadas en espacios de pentagrama.

El ángulo de rotación y las coordenadas del punto de rotación se deben determinar por ensayo y error.

Solamente en ciertas ocasiones es útil la rotación de objetos de presentación; el ejemplo siguiente muestra una situación en que puede serlo:
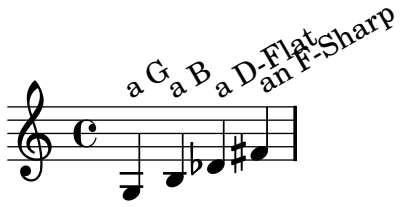
```
g4\< e' d' f\!
\override Hairpin #'rotation = #'(20 -1 0)
g,,4\< e' d' f\!
```



Rotación de elementos de marcado

Todos los textos de marcado se pueden rotar para que se dispongan en cualquier ángulo predeciéndolos de la instrucción `\rotate`. La instrucción acepta dos argumentos: el ángulo de rotación en grados en sentido antihorario, y el texto que rotar. Los límites que ocupa el texto no se rotan: toman su valor a partir de los extremos de las coordenadas x e y del texto rotado. En el ejemplo siguiente la propiedad `outside-staff-priority` del texto se establece a `#f` para desactivar la evitación automática de colisiones, lo que empuja al texto a una posición muy alta.

```
\override TextScript #'outside-staff-priority = ##f
g4^\markup { \rotate #30 "a G" }
b^\markup { \rotate #30 "a B" }
des^\markup { \rotate #30 "a D-Flat" }
fis^\markup { \rotate #30 "an F-Sharp" }
```



5.5 Trucos avanzados

Esta sección trata sobre distintos enfoques en la realización de ajustes finos a la apariencia de la partitura impresa.

Véase también

Manual de aprendizaje: Sección “Tweaking output” in *Manual de Aprendizaje*, Sección “Other sources of information” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.2 [Explicación del Manual de referencia de funcionamiento interno], página 397, Sección 5.3 [Modificar las propiedades], página 400, Capítulo 6 [Interfaces para programadores], página 429.

Archivos de inicio: ‘`scm/define-grobs.scm`’.

Fragmentos de código: Sección “Tweaks and overrides” in *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “All layout objects” in *Referencia de Funcionamiento Interno*.

5.5.1 Alineación de objetos

Los objetos gráficos que soportan el interface `self-alignment-interface` y/o el `side-position-interface` se pueden alinear contra un objeto colocado previamente, de diversas maneras. Para ver una lista de estos objetos, consulte Sección “self-alignment-interface” in *Referencia de Funcionamiento Interno* y Sección “side-position-interface” in *Referencia de Funcionamiento Interno*.

Todos los objetos gráficos tienen un punto de referencia, una extensión horizontal y una extensión vertical. La extensión horizontal es una pareja de números que dan los desplazamientos a partir del punto de referencia de los bordes izquierdo y derecho, siendo negativos los desplazamientos hacia la izquierda. La extensión vertical es una pareja de números que dan el desplazamiento a partir del punto de referencia hasta los bordes inferior y superior, siendo negativos los deplazamientos hacia abajo.

La posición de un objeto sobre el pentagrama viene dada por los valores de las propiedades `X-offset` e `Y-offset`. El valor de `X-offset` da el desplazamiento desde la coordenada x del punto de referencia del objeto padre, y el valor de `Y-offset` da el desplazamiento a partir de la línea central del pentagrama. Los valores de `X-offset` y `Y-offset` se pueden establecer directamente o se puede dejar que se calculen por parte de procedimientos para conseguir una alineación con el objeto padre de distintas maneras.

Nota: Muchos objetos tienen consideraciones de posicionamiento especiales que hacen que se ignore o se modifique cualquier ajuste realizado a `X-offset` o a `Y-offset`, a pesar de que el objeto contemple el interface `self-alignment-interface`.

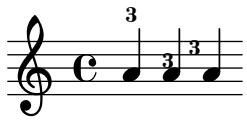
Por ejemplo, una alteración accidental se puede reposicionar verticalmente estableciendo `Y-offset` pero los cambios a `X-offset` no tienen ningún efecto.

Las letras de ensayo se pueden alinear con objetos divisibles como líneas divisorias, claves, armaduras e indicaciones de compás. Hay propiedades especiales que se encuentran en `break-aligned-interface` para el posicionamiento de las letras de ensayo sobre dichos objetos.

Establecer directamente `X-offset` y `Y-offset`

Se pueden dar valores numéricos a las propiedades `X-offset` y `Y-offset` de muchos objetos. El ejemplo siguiente muestra tres notas con una digitación en su posición predeterminada y con los valores `X-offset` y `Y-offset` modificados.

```
a-3
a
-\tweak #'X-offset #0
-\tweak #'Y-offset #0
-3
a
-\tweak #'X-offset #-1
-\tweak #'Y-offset #1
-3
```



Uso del `side-position-interface`

Un objeto que contempla el `side-position-interface` se puede colocar junto a su objeto padre de forma que los bordes especificados de los dos objetos se toquen. El objeto se puede situar encima, debajo, a la derecha o a la izquierda del objeto padre. El padre no se puede especificar; ciene determinado por el orden de los elementos en el flujo de entrada. Casi todos los objetos tienen la cabeza de la nota asociada como padre.

Los valores de las propiedades `side-axis` y `direction` determinan dónde colocar el objeto, como sigue:

side-axis **propiedad**	direction **propiedad**	**colocación**
0	-1	izquierda
0	1	derecha
1	-1	debajo
1	1	encima

Si side-axis es 0, X-offset se debe establecer al procedimiento ly:side-position-interface::x-aligned-side. Este procedimiento devuelve el calor correcto de X-offset para situar el objeto al lado izquierdo o derecho del padre de acuerdo con el valor de direction.

Si side-axis es 1, Y-offset se debe establecer al procedimiento ly:side-position-interface::y-aligned-side. Este procedimiento devuelve el valor correcto de Y-offset para situar el objeto encima o debajo del padre de acuerdo con el valor de direction.

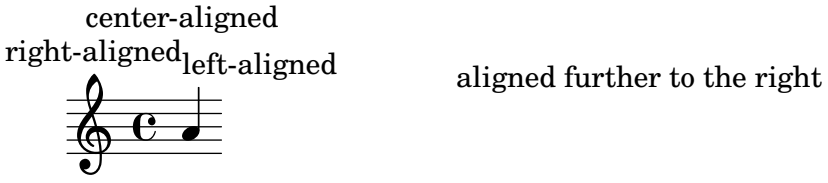
Uso del self-alignment-interface

Auto-alineación horizontal de los objetos

La alineación horizontal de un objeto que contempla el interface self-alignment-interface está controlada por el valor de la propiedad self-alignment-X, siempre y cuando la propiedad X-offset de este objeto esté establecida a ly:self-alignment-interface::x-aligned-on-self. Se le puede dar a self-alignment-X cualquier valor real, en unidades de la mitad de la extensión X total del objeto. Los valores negativos mueven el objeto a la derecha, los positivos hacia la izquierda. Un valor de 0 centra el objeto sobre el punto de referencia de su padre, un valor de -1 alinea el borde izquierdo del objeto sobre el punto de referencia de su padre, y un valor de 1 alinea el borde derecho del objeto sobre el punto de referencia de su padre. Se pueden usar los símbolos LEFT, CENTER y RIGHT en sustitución de los valores -1, 0, 1, respectivamente.

Normalmente se usaría la instrucción \override para modificar el valor de self-alignment-X, pero se puede usar la instrucción \tweak para alinear varias anotaciones por separado sobre una sola nota:

```
a'
-\tweak #'self-alignment-X #-1
^"left-aligned"
-\tweak #'self-alignment-X #0
^"center-aligned"
-\tweak #'self-alignment-X #RIGHT
^"right-aligned"
-\tweak #'self-alignment-X #-2.5
^"aligned further to the right"
```



Auto-alineación vertical de los objetos

Los objetos se pueden alinear verticalmente en una forma análoga a la alineación horizontal si la propiedad Y-offset está establecida a ly:self-alignment-interface::y-aligned-on-self. Sin embargo, a menudo se encuentran implicados otros mecanismos en la alineación vertical: el valor de Y-offset es tan sólo una variable que se tiene en cuenta. Esto puede hacer que
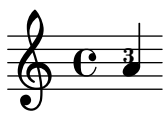
ajustar el valor de ciertos objetos sea una tarea dificultosa. Las unidades son sólo la mitad de las dimensiones verticales del objeto, que suele ser bastante pequeño, por lo que pueden requerirse números bastante grandes. Un valor de `-1` alinea el borde inferior del objeto con el punto de referencia del objeto padre, un valor de `0` alinea el centro del objeto con el punto de referencia del padre, y un valor de `1` alinea el borde superior del objeto con el punto de referencia del padre. Se pueden usar los símbolos `DOWN`, `CENTER`, `UP` en sustitución de `-1, 0, 1` respectivamente.

Auto-alineación de objetos en las dos direcciones

Estableciendo tanto `X-offset` como `Y-offset`, se puede alinear un objeto en las dos direcciones simultáneamente.

El ejemplo siguiente muestra cómo ajustar una digitación de forma que se acerque a la cabeza de la nota.

```
a
-\tweak #'self-alignment-X #0.5  % desplazar a la izquierda
-\tweak #'Y-offset #ly:self-alignment-interface::y-aligned-on-self
-\tweak #'self-alignment-Y #-1  % desplazar hacia arriba
-3  % dedo tercero
```

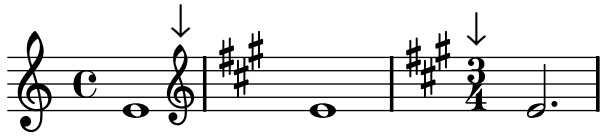


Uso del `break-alignable-interface`

Las letras de ensayo se pueden alinear con objetos de notación distintos a las barras de compás. Estos objetos son `ambitus`, `breathing-sign`, `clef`, `custos`, `staff-bar`, `left-edge`, `key-cancellation`, `key-signature` y `time-signature`.

De forma predeterminada, las letras de ensayo y los números de compás se centran horizontalmente sobre el objeto:

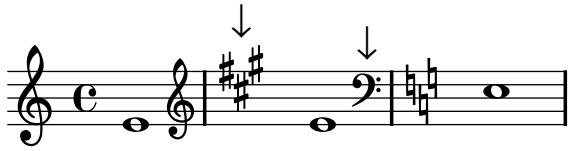
```
e1
% la RehearsalMark se centra sobre la clave
\override Score.RehearsalMark #'break-align-symbols = #'(clef)
\key a \major
\clef treble
\mark ""
e
% la RehearsalMark se centra sobre el compás
\override Score.RehearsalMark #'break-align-symbols = #'(time-signature)
\key a \major
\clef treble
\time 3/4
\mark ""
e2.
```



Se puede especificar una lista de posibles objetos para la alineación. Si algunos de los objetos son invisibles en ese punto debido al valor de `break-visibility` o a valores de visibilidad explícitos para las armaduras y las claves, la letra de ensayo o número de compás se alinean con

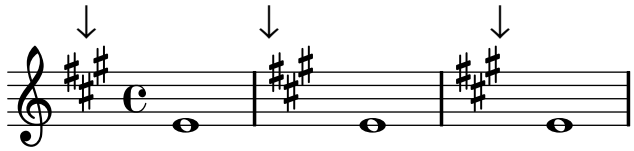
el primer objeto de la lista que sea visible. Si ningún objeto de la lista es visible, el objeto se alinea con la línea divisoria. Si la línea divisoria es invisible, el objeto se alinea con el punto en el que se encontraría la línea divisoria.

```
e1
% la RehearsalMark se centra sobre la armadura
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark ""
e
% la RehearsalMark se centra sobre la clave
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature clef)
\key a \minor
\clef bass
\mark ""
e,
```



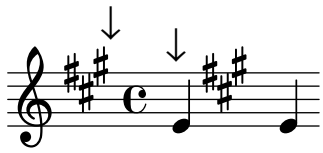
La alineación de la letra de ensayo con relación al objeto de notación se puede cambiar, como se ve en el ejemplo siguiente. En una partitura con varios pentagramas, este ajuste se debe hacer para todos los pentagramas.

```
% la RehearsalMark se centra sobre la armadura
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature)
\key a \major
\clef treble
\time 4/4
\mark ""
e1
% la RehearsalMark se alinea por la izquierda con la armadura
\once \override Score.KeySignature #'break-align-anchor-alignment = #LEFT
\mark ""
\key a \major
e
% la RehearsalMark se alinea por la derecha con la armadura
\once \override Score.KeySignature #'break-align-anchor-alignment = #RIGHT
\key a \major
\mark ""
e
```



La letra de ensayo también se puede desplazar al borde derecho o izquierdo en una medida arbitraria. Las unidades son espacios de pentagrama:

```
% la RehearsalMark se alinea por la izquierda con la armadura
% y se desplaza 3.5 espacios a la derecha
\override Score.RehearsalMark #'break-align-symbols = #'(key-signature)
\once \override Score.KeySignature #'break-align-anchor = #3.5
\key a \major
\mark ""
e
% la RehearsalMark se alinea por la izquierda con la armadura
% y se desplaza 2 espacios a la izquierda
\once \override Score.KeySignature #'break-align-anchor = #-2
\key a \major
\mark ""
e
```



5.5.2 Agrupación vertical de objetos gráficos

Los grobs (objetos gráficos) `VerticalAlignment` y `VerticalAxisGroup` trabajan de manera coordinada. `VerticalAxisGroup` agrupa distintos grobs como `Staff`, `Lyrics`, etc. Después, `VerticalAlignment` alinea los distintos grobs agrupados previamente por `VerticalAxisGroup`. Normalmente sólo existe un `VerticalAlignment` por cada partitura, pero cada `Staff`, `Lyrics`, etc. tiene su propio `VerticalAxisGroup`.

5.5.3 Modificación de los sellos

Todos los objetos de presentación tienen una propiedad `stencil` que es parte del `grob-interface`. De forma predeterminada, esta propiedad suele estar establecida a una función específica del objeto que está hecha a medida para disponer el símbolo que lo representa en la salida. Por ejemplo, el ajuste estándar para la propiedad `stencil` del objeto `MultiMeasureRest` es `ly:multi-measure-rest::print`.
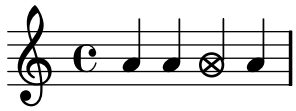
El símbolo estándar de cualquier objeto se puede sustituir modificando la propiedad `stencil` para que haga referencia a un procedimiento diferente escrito especialmente. Esto requiere un alto nivel de conocimiento del funcionamiento interno de LilyPond, pero hay una forma más fácil que a menudo puede dar resultados adecuados.

El procedimiento es establecer la propiedad `stencil` al procedimiento que imprime texto (`ly:text-interface::print`) y añadir una propiedad `text` al objeto ajustada para que contenga el texto de marcado que produce el símbolo requerido. Debido a la flexibilidad del marcado, se pueden conseguir muchas cosas; en particular, consulte [Notación gráfica dentro de elementos de marcado], página 180.

El ejemplo siguiente muestra esto cambiando el símbolo de la cabeza de la nota a unas aspas dentro de una circunferencia.

```
X_O = {
  \once \override NoteHead  #'stencil = #ly:text-interface::print
  \once \override NoteHead #'text = \markup {
    \combine
      \halign #-0.7 \draw-circle #0.85 #0.2 ##f
      \musicglyph #"noteheads.s2cross"
  }
```

```
}
\relative c'' {
  a a \X_O a a
}
```



Cualquiera de los glifos de la fuente tipográfica Feta se puede pasar a la instrucción de marcado `\musicglyph`: véase Sección B.6 [La tipografía Feta], página 461.

Véase también

Referencia de la notación: [Notación gráfica dentro de elementos de marcado], página 180, Sección 1.8.2 [Formatear el texto], página 173, Sección B.8 [Text markup commands], página 476, Sección B.6 [La tipografía Feta], página 461.

5.5.4 Modificación de las formas
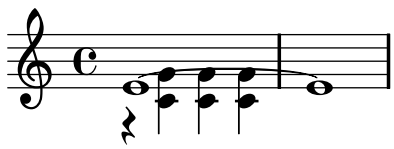
Modificación de ligaduras de unión y de expresión

Las ligaduras de unión, de expresión y de fraseo se trazan como curvas de Bézier de tercer orden. Si la forma de la ligadura calculada automáticamente no resulta óptima, se puede modificar su forma manualmente mediante la especificación explícita de los cuatro puntos de control necesarios para definir una curva de Bézier de tercer orden.

Las curvas Bézier de tercer orden o cúbicas están definidas por cuatro puntos de control. El primer y cuarto puntos de control son exactamente los puntos extremos de comienzo y de final de la curva. Los dos puntos de contros intermedios definen la forma. Se pueden encontrar en la web animaciones que muestran cómo se traza la curva, pero la descripción siguiente puede ser de ayuda. La curva comienza a partir del primer punto de control dirigiéndose directamente hacia el segundo, curvándose progresivamente para dirigirse hacia el tercero y continuando la curva hacia el cuarto, llegando a éste en viaje directo desde el tercer punto de control. La curva está contenida enteramente dentro del cuadrilátero definido por los cuatro puntos de control.

He aquí un ejemplo de un caso en que la ligadura no es óptima, y donde `\tieDown` no sirve de ayuda.

```
<<
  { e1 ~ e }
\\
  { r4 <g c,> <g c,> <g c,> }
>>
```



Una forma de mejorar esta ligadura es modificar manualmente sus puntos de control como sigue.
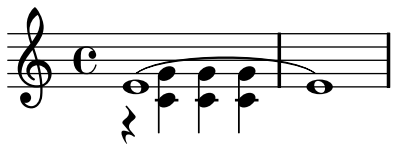
Las coordenadas de los puntos de control de Bézier se especifican en unidades de espacios de pentagrama. La coordenada X está en relación con el punto de referencia de la nota a la que está unida la ligadura, y la coordenada Y está en relación con la línea central del pentagrama. Las coordenadas se introducen como una lista de cuatro parejas de números decimales (reales).

Un enfoque es estimar las coordenadas de los dos puntos extremos, y luego tratar de adivinar los dos puntos intermedios. Los valores óptimos se encuentran por ensayo y error.

Es útil recordar que una curva simétrica necesita puntos de contros simétricos, y que las curvas de Bézier tienen la útil propiedad de que las transformaciones de la curva tales como la traslación, rotación y escalado se pueden obtener aplicando la misma transformación a los puntos de control de la curva.

Para el ejemplo anterior, la sobreescritura siguiente da una ligadura satisfactoria:

```
<<
  \once \override Tie
    #'control-points = #'((1 . -1) (3 . 0.6) (12.5 . 0.6) (14.5 . -1))
  { e1 ~ e1 }
\\
  { r4 <g c,> <g c,> <g c,>4  }
>>
```



Advertencias y problemas conocidos

No es posible modificar la forma de las ligaduras de unión o de expresión cambiando la propiedad `control-points` si hay más de una en el mismo mom ento musical, ni siquiera usando la instrucción `\tweak`.

6 Interfaces para programadores

Se pueden realizar trucos avanzados mediante el uso de Scheme. Si no está familizarizado con Scheme, le conviene leer nuestro tutorial de Scheme, Sección “Scheme tutorial” in *Manual de Aprendizaje*.

6.1 Funciones musicales

Esta sección trata sobre cómo crear funciones musicales dentro de LilyPond.

6.1.1 Panorámica de las funciones musicales

Es fácil hacer una función que sustituya a una variable en código de LilyPond. La forma general de estas funciones es:

```
function =
#(define-music-function (parser location var1 var2...vari... )
                        (var1-type? var2-type?...vari-type?...)
  #{
    ...música...
  #})
```

donde

vari	*i*-ésima variable
vari-type?	tipo de la *i*-ésima variable
...música...	entrada normal de LilyPond, usando las variables como `#$var1`, etc.

Los siguientes tipos de entrada se pueden usar como variables en una función musical. Esta lista no es exhaustiva; consulte otros lugares de la documentación específica de Scheme para ver otros tipos de variables.

Tipo de entrada	**notación de** *vari-type?*
Entero	`integer?`
Flotante (número decimal)	`number?`
Cadena de texto	`string?`
Marcado	`markup?`
Expresión musical	`ly:music?`
Pareja de variables	`pair?`

Los argumentos `parser` y `location` son obligatorios, y se usan en ciertas situaciones avanzadas. El argumento `parser` se usa para tener acceso al valor de otra variable de LilyPond. El argumento `location` se usa para establecer el ‘origen’ de la expresión musical que construye la función musical, de forma que en caso de producirse un error de sintaxis LilyPond pueda informar al usuario de un lugar adecuado donde buscar en el archivo de entrada.
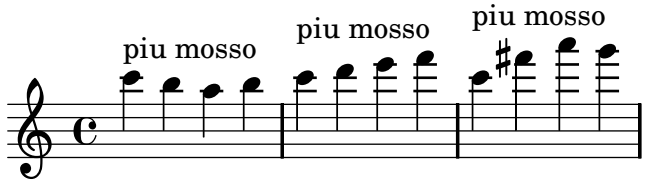
6.1.2 Funciones de sustitución sencillas

He aquí un ejemplo sencillo:

```
textoRelleno = #(define-music-function (parser location padding) (number?)
  #{
    \once \override TextScript #'padding = #$padding
  #})

\relative c''' {
  c4^"piu mosso" b a b
```
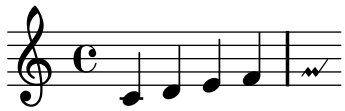
```
  \textoRelleno #1.8
  c4^"piu mosso" d e f
  \textoRelleno #2.6
  c4^"piu mosso" fis a g
}
```



También se pueden sustituir las expresiones musicales:

```
notaCustos = #(define-music-function (parser location note)
                                     (ly:music?)
  #{
    \once \override Voice.NoteHead #'stencil =
      #ly:text-interface::print
    \once \override Voice.NoteHead #'text =
      \markup \musicglyph #"custodes.mensural.u0"
    \once \override Voice.Stem #'stencil = ##f
    $note
  #})
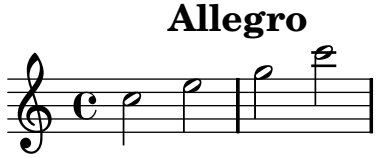
{ c' d' e' f' \notaCustos g' }
```



Se pueden usar más de una variable:

```
marcaDeTempo = #(define-music-function (parser location padding marktext)
                                       (number? string?)
#{
  \once \override Score . RehearsalMark #'padding = $padding
  \once \override Score . RehearsalMark #'extra-spacing-width = #'(+inf.0 . -inf.0)
  \mark \markup { \bold $marktext }
#})

\relative c'' {
c2 e
\marcaDeTempo #3.0 #"Allegro"
g c
}
```



6.1.3 Funciones de sustitutión en parejas

Algunas instrucciones `\override` requieren un par de números (llamados en Scheme una `célula cons`). Para pasar estos números a una función, usamos una variable `pair?` o bien insertamos el `cons` en la función musical.

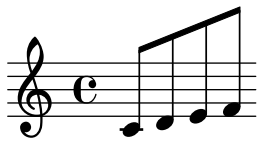
```
manualBeam =
#(define-music-function (parser location beg-end)
                        (pair?)
#{
  \once \override Beam #'positions = #$beg-end
#})

\relative {
  \manualBeam #'(3 . 6) c8 d e f
}
```

o bien

```
barraManual =
#(define-music-function (parser location beg end)
                        (number? number?)
#{
  \once \override Beam #'positions = #(cons $beg $end)
#})

\relative {
  \barraManual #3 #6 c8 d e f
}
```
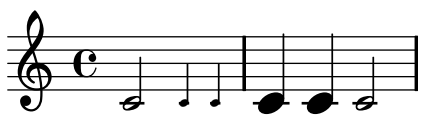


6.1.4 Matemáticas dentro de las funciones

Las funciones musicales pueden contar con programación de Scheme además de la simple sustitución:

```
varSi = #(define-music-function (parser location mag) (number?)
  #{ \override Stem #'length = #$(* 7.0 mag)
     \override NoteHead #'font-size =
       #$(inexact->exact (* (/ 6.0 (log 2.0)) (log mag))) #})

varNo = {
  \revert Stem #'length
  \revert NoteHead #'font-size
}
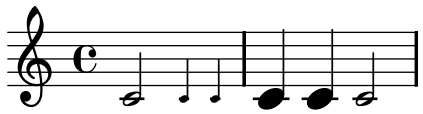
{ c'2 \varSi #0.5 c'4 c'
  \varSi #1.5 c' c' \varNo c'2 }
```



Este ejemplo se puede reescribir de forma que pase expresiones musicales:

```
variante = #(define-music-function (parser location mag music) (number? ly:music?)
  #{ \override Stem #'length = #$(* 7.0 mag)
     \override NoteHead #'font-size =
       #$(inexact->exact (* (/ 6.0 (log 2.0)) (log mag)))
     $music
     \revert Stem #'length
     \revert NoteHead #'font-size #})

{ c'2 \variante #0.5 {c'4 c'}
  \variante #1.5 {c' c'} c'2 }
```



6.1.5 Funciones vacías

Una función musical debe devolver una expresión musical, per a veces podemos necesitar una función en la que no hay música en juego (como la desactivación de la funcionalidad Apuntar y Pulsar). Para hacerlo, devolvemos una expresión musical `void` (vacía).

Este es el motivo por el que la forma que se devuelve es `(make-music ...)`. Con el valor de la propiedad `'void` establecido a `#t`, le decimos al analizador que descarte la expresión musical devuelta. así, la parte importante de la función musical vacía es el proceso realizado por la función, no la expresión musical que se devuelve.

```
noPointAndClick =
#(define-music-function (parser location) ()
   (ly:set-option 'point-and-click #f)
   (make-music 'SequentialMusic 'void #t))
...
\noPointAndClick   % desactivar la funcionalidad Apuntar y Pulsar.
```

6.1.6 Funciones sin argumentos

En casi todos los casos, una función sin argumentos se debe escribir con una variable:

```
dolce = \markup{ \italic \bold dolce }
```

Sin embargo, en raras ocasiones puede ser de utilidad crear una función musical sin argumentos:

```
displayBarNum =
#(define-music-function (parser location) ()
   (if (eq? #t (ly:get-option 'display-bar-numbers))
       #{ \once \override Score.BarNumber #'break-visibility = ##f #}
       #{#}))
```

Para la imresión real de los números de compás donde se llama a esta función, invoque a `lilypond` con

```
lilypond -d display-bar-numbers ARCHIVO.ly
```

6.1.7 Pranorámica de las funciones musicales disponibles

Las siguientes instrucciones son funciones musicales:

`acciaccatura` - *music* (music)
(undocumented; fixme)

`addChordShape` - *key-symbol* (symbol) *tuning* (pair) *shape-definition* (unknown)
: Add chord shape `shape-definition` to the `chord-shape-table` hash with the key `(cons key-symbol tuning)`.

`addInstrumentDefinition` - *name* (string) *lst* (list)
: (undocumented; fixme)

`addQuote` - *name* (string) *music* (music)
: Add a piece of music to be quoted

`afterGrace` - *main* (music) *grace* (music)
: (undocumented; fixme)

`allowPageTurn`
: Allow a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

`applyContext` - *proc* (procedure)
: (undocumented; fixme)

`applyMusic` - *func* (procedure) *music* (music)
: (undocumented; fixme)

`applyOutput` - *ctx* (symbol) *proc* (procedure)
: (undocumented; fixme)

`appoggiatura` - *music* (music)
: (undocumented; fixme)

`assertBeamQuant` - *l* (pair) *r* (pair)
: (undocumented; fixme)

`assertBeamSlope` - *comp* (procedure)
: (undocumented; fixme)

`autochange` - *music* (music)
: (undocumented; fixme)

`balloonGrobText` - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)
: (undocumented; fixme)

`balloonText` - *offset* (pair of numbers) *text* (markup)
: (undocumented; fixme)

`bar` - *type* (string)
: (undocumented; fixme)

`barNumberCheck` - *n* (integer)
: (undocumented; fixme)

`bendAfter` - *delta* (unknown)
: (undocumented; fixme)

`breathe` (undocumented; fixme)

`clef` - *type* (string)
: Set the current clef.

`cueDuring` - *what* (string) *dir* (direction) *main-music* (music)
: (undocumented; fixme)

`displayLilyMusic` - *music* (music)
: (undocumented; fixme)

`displayMusic` - *music* (music)
(undocumented; fixme)

`endSpanners` - *music* (music)
(undocumented; fixme)

`featherDurations` - *factor* (moment) *argument* (music)
Rearrange durations in ARGUMENT so there is an acceleration/deceleration.

`grace` - *music* (music)
(undocumented; fixme)

`includePageLayoutFile`
If page breaks and tweak dump is not asked, and the file <basename>-page-layout.ly exists, include it.

`instrumentSwitch` - *name* (string)
(undocumented; fixme)

`keepWithTag` - *tag* (symbol) *music* (music)
(undocumented; fixme)

`killCues` - *music* (music)
(undocumented; fixme)

`label` - *label* (symbol)
Place a bookmarking label, either at top-level or inside music.

`makeClusters` - *arg* (music)
(undocumented; fixme)

`musicMap` - *proc* (procedure) *mus* (music)
(undocumented; fixme)

`noPageBreak`
Forbid a page break. May be used at toplevel (ie between scores or markups), or inside a score.

`noPageTurn`
Forbid a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

`octaveCheck` - *pitch-note* (music)
octave check

`oldaddlyrics` - *music* (music) *lyrics* (music)
(undocumented; fixme)

`ottava` - *octave* (number)
set the octavation

`overrideProperty` - *name* (string) *property* (symbol) *value* (any type)
Set *property* to *value* in all grobs named *name*. The *name* argument is a string of the form `"Context.GrobName"` or `"GrobName"`

`pageBreak`
Force a page break. May be used at toplevel (ie between scores or markups), or inside a score.

`pageTurn` Force a page turn between two scores or top-level markups.

`parallelMusic` - *voice-ids* (list) *music* (music)
Define parallel music sequences, separated by '`|`' (bar check signs), and assign them to the identifiers provided in *voice-ids*.

voice-ids: a list of music identifiers (symbols containing only letters)

music: a music sequence, containing BarChecks as limiting expressions.

Example:

```
\parallelMusic #'(A B C) {
  c c | d d | e e |
  d d | e e | f f |
}
<==>
  A = { c c | d d | }
  B = { d d | e e | }
  C = { e e | f f | }
```

`parenthesize` - *arg* (music)
Tag *arg* to be parenthesized.

`partcombine` - *part1* (music) *part2* (music)
(undocumented; fixme)

`pitchedTrill` - *main-note* (music) *secondary-note* (music)
(undocumented; fixme)

`pointAndClickOff`
(undocumented; fixme)

`pointAndClickOn`
(undocumented; fixme)

`quoteDuring` - *what* (string) *main-music* (music)
(undocumented; fixme)

`removeWithTag` - *tag* (symbol) *music* (music)
(undocumented; fixme)

`resetRelativeOctave` - *reference-note* (music)
Set the octave inside a \relative section.

`rightHandFinger` - *finger* (number or string)
Define a StrokeFingerEvent

`scaleDurations` - *fraction* (pair of numbers) *music* (music)
(undocumented; fixme)

`scoreTweak` - *name* (string)
Include the score tweak, if exists.

`shiftDurations` - *dur* (integer) *dots* (integer) *arg* (music)
`spacingTweaks` - *parameters* (list)
Set the system stretch, by reading the 'system-stretch property of the 'parameters' assoc list.

`storePredefinedDiagram` - *chord* (music) *tuning* (pair) *diagram-definition* (unknown)
Add predefined fret diagram defined by `diagram-definition` for the chord pitches `chord` and the stringTuning `tuning`.

`tag` - *tag* (symbol) *arg* (music)
Add *tag* to the `tags` property of *arg*.

`tocItem` - *text* (markup)
: Add a line to the table of content, using the `tocItemMarkup` paper variable markup

`transposedCueDuring` - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)
: Insert notes from the part *what* into a voice called `cue`, using the transposition defined by *pitch-note*. This happens simultaneously with *main-music*, which is usually a rest. The argument *dir* determines whether the cue notes should be notated as a first or second voice.

`transposition` - *pitch-note* (music)
: Set instrument transposition

`tweak` - *sym* (symbol) *val* (any type) *arg* (music)
: Add `sym . val` to the `tweaks` property of *arg*.

`unfoldRepeats` - *music* (music)
: (undocumented; fixme)

`withMusicProperty` - *sym* (symbol) *val* (any type) *music* (music)
: Set *sym* to *val* in *music*.

6.2 Interfaces para el programador

Esta sección contiene información sobre cómo mezclar LilyPond y Scheme.

6.2.1 Variables de entrada y Scheme

El formato de entrada contempla la noción de variables: en el ejemplo siguiente, se asigna una expresión musical a una variable con el nombre `traLaLa`.

```
traLaLa = { c'4 d'4 }
```

También existe una forma de ámbito léxico: en el ejemplo siguiente, el bloque `\layout` también contiene una variable `traLaLa`, que es independiente de la `\traLaLa` exterior.

```
traLaLa = { c'4 d'4 }
\layout { traLaLa = 1.0 }
```

De hecho, cada archivo de entrada es un ámbito léxico, y todos los bloques `\header`, `\midi` y `\layout` son ámbitos anidados dentro de dicho ámbito de nivel superior.

Tanto el ámbito léxico como las variables están implementados en el sistema de módulos GUILE. Se adjunta un módulo anónimo de Scheme a cada ámbito. Una asignación de la forma

```
traLaLa = { c'4 d'4 }
```

se convierte internamente a una definición de Scheme

```
(define traLaLa Scheme value of '... ')
```

Esto supone que las variables de entrada y las variables de Scheme se pueden intermezclar con libertad. En el ejemplo siguiente, se almacena un fragmento musical en la variable `traLaLa`, y se dupplica utilizando Scheme. El resultado se importa en un bloque `\score` por medio de una segunda variable `twice`:
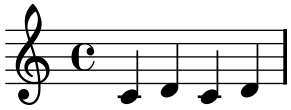
```
traLaLa = { c'4 d'4 }

%% acción vacía para la predicción del analizador sintáctico
#(display "this needs to be here, sorry!")

#(define newLa (map ly:music-deep-copy
  (list traLaLa traLaLa)))
#(define twice
```

```
  (make-sequential-music newLa))

{ \twice }
```



En este ejemplo, la asignación se produce después de que el analizador sintáctico ha verificado que no ocurre nada interesante después de `traLaLa = { ... }`. Sin el argumento mudo del ejemplo, la definición `newLa` se ejecuta antes de que se defina `traLaLa`, conduciendo a un error de sintaxis.

El ejemplo anterior muestra cómo 'exportar' expresiones musicales desde la entrada hasta el intérprete de Scheme. También es posible lo contrario. Envolviendo un valor de Scheme en la función `ly:export`, un valor de Scheme se interpreta como si hubiera sido introducido en sintaxis de LilyPond. En vez de definir `\twice`, el ejemplo anterior podría también haberse escrito como

```
...
{ #(ly:export (make-sequential-music (list newLa))) }
```

El sódigo de Scheme se evalúa tan pronto como el analizador sintáctico lo encuentra. Para definir código de Scheme en un macro (para llamarlo con posterioridad), use Sección 6.1.5 [Funciones vacías], página 432, o bien

```
#(define (nopc)
  (ly:set-option 'point-and-click #f))

...
#(nopc)
{ c'4 }
```

Advertencias y problemas conocidos

No es posible mezclar variables de Scheme y de LilyPond con la opción `--safe`.

6.2.2 Representación interna de la música

Cuando se analiza sintácticamente una expresión musical, se convierte en un conjunto de objetos musicales de Scheme. La propiedad que define a un objeto musical es que tiene una cierta duración. El tiempo es un número racional que mide la longitud de un fragmento de música en unidades del valor de una redonda.

Un objeto musical tiene tres clases de tipos:

- nombre musical: cada expresión musical tiene un nombre. Por ejemplo, una nota conduce a un evento Sección "NoteEvent" in *Referencia de Funcionamiento Interno*, y `\simultaneous` conduce a Sección "SimultaneousMusic" in *Referencia de Funcionamiento Interno*. Hay una lista de todas las expresiones que están disponibles en el Manual de referencia de funcionamiento interno, bajo Sección "Music expressions" in *Referencia de Funcionamiento Interno*.
- 'typo' o intterface: cada nombre de música tiene varios 'tipos' o interfaces, por ejemplo una nota es un `event`, pero también es un `note-event`, un `rhythmic-event` y un `melodic-event`. Todas las clases musicales se encuentran relacionadas en la Referencia de funcionamiento interno bajo Sección "Music classes" in *Referencia de Funcionamiento Interno*.
- Objeto de C++: cada objeto musical está representado por un objeto de la clase de C++ `Music`.

La información real de una expresión musical se almacena en forma de propiedades. Por ejemplo, un evento Sección “NoteEvent” in *Referencia de Funcionamiento Interno* tiene propiedades `pitch` y `duration` que almacenan la altura y duración de la nota. Hay una lista completa de las propiedades que están disponibles en la Referencia de funcionamiento interno, bajo Sección “Music properties” in *Referencia de Funcionamiento Interno*.

Una expresión musical compuesta es un objeto musical que contiene otros objetos musicales en sus propiedades. Se puede almacernar una lista de objetos en la propiedad `elements` de un objeto musical, o un solo objeto musical ‘hijo’ en la propiedad `element`. Por ejemplo, Sección “SequentialMusic” in *Referencia de Funcionamiento Interno* tiene sus hijos en `elements`, y Sección “GraceMusic” in *Referencia de Funcionamiento Interno* tiene su elemento único en `element`. El cuerpo de una repetición se almacena en la propiedad `element` de Sección “RepeatedMusic” in *Referencia de Funcionamiento Interno*, y las alternativas en `elements`.

6.3 Construcción de funciones complejas

Esta sección explica cómo reunir la información necesaria para crear funciones musicales complejas.

6.3.1 Presentación de expresiones musicales

Si se está escribiendo una función musical puede ser muy instructivo examinar cómo se almacena internamente una expresión musical. Esto se puede hacer con la función musical `\displayMusic`:

```
{
  \displayMusic { c'4\f }
}
```

imprime lo siguiente:

```
(make-music
  'SequentialMusic
  'elements
  (list (make-music
          'EventChord
          'elements
          (list (make-music
                  'NoteEvent
                  'duration
                  (ly:make-duration 2 0 1 1)
                  'pitch
                  (ly:make-pitch 0 0 0))
                (make-music
                  'AbsoluteDynamicEvent
                  'text
                  "f")))))
```

De forma predeterminada, LilyPond imprime estos mensajes en la consola junto al resto de los mensajes. Para discernir entre estos mensajes y guardar el resultado de `\display{MATERIAL}`, redirija la salida hacia un archivo.

```
lilypond archivo.ly >resultado.txt
```

Con la aplicación de un poco de formato, la inforamción anterior es fácil de leer:

```
(make-music 'SequentialMusic
  'elements (list (make-music 'EventChord
                    'elements (list (make-music 'NoteEvent
                                      'duration (ly:make-duration 2 0 1 1)
```

```
                                        'pitch (ly:make-pitch 0 0 0))
                                      (make-music 'AbsoluteDynamicEvent
                                        'text "f")))))
```

Una secuencia musical `{ ... }` tiene el nombre `SequentialMusic`, y sus expresiones internas se almacenan como una lista en su propiedad `'elements`. Una nota se representa como una expresión `EventChord` que contiene un objeto `NoteEvent` (que almacena las propiedades de duración y altura) y cualquier otra información adicional (en este caso, un evento `AbsoluteDynamicEvent` con una porpiedad de texto `"f"`.

6.3.2 Propiedades de la música

El objeto `NoteEvent` es el primer objeto de la propiedad `'elements` de `someNote`.

```
unaNota = c'
\displayMusic \unaNota
===>
(make-music
  'EventChord
  'elements
  (list (make-music
          'NoteEvent
          'duration
          (ly:make-duration 2 0 1 1)
          'pitch
          (ly:make-pitch 0 0 0))))
```

La función `display-scheme-music` es la función utilizada por `\displayMusic` para imprimir la representación de Scheme de una expresión musical.

```
#(display-scheme-music (first (ly:music-property unaNota 'elements)))
===>
(make-music
  'NoteEvent
  'duration
  (ly:make-duration 2 0 1 1)
  'pitch
  (ly:make-pitch 0 0 0))
```

Después se accede a la altura de la nota a través de la propiedad `'pitch` del objeto `NoteEvent`:

```
#(display-scheme-music
   (ly:music-property (first (ly:music-property unaNota 'elements))
                      'pitch))
===>
(ly:make-pitch 0 0 0)
```

La altura de la nota se puede cambiar estableciendo el valor de esta propiedad 'pitch:

```
#(set! (ly:music-property (first (ly:music-property unaNota 'elements))
                          'pitch)
       (ly:make-pitch 0 1 0)) ;; fijar la altura a d'.
\displayLilyMusic \unaNota
===>
d'
```

6.3.3 Doblar una nota con ligaduras (ejemplo)

Supongamos que queremos crear una función que traduce una entrada como `a` a algo como `a( a)`. Empezamos examinando la representación interna de la música con la que queremos terminar.

```
\displayMusic{ a'( a') }
===>
(make-music
  'SequentialMusic
  'elements
  (list (make-music
          'EventChord
          'elements
          (list (make-music
                  'NoteEvent
                  'duration
                  (ly:make-duration 2 0 1 1)
                  'pitch
                  (ly:make-pitch 0 5 0))
                (make-music
                  'SlurEvent
                  'span-direction
                  -1)))
        (make-music
          'EventChord
          'elements
          (list (make-music
                  'NoteEvent
                  'duration
                  (ly:make-duration 2 0 1 1)
                  'pitch
                  (ly:make-pitch 0 5 0))
                (make-music
                  'SlurEvent
                  'span-direction
                  1)))))
```

Las malas noticias son que las espresiones `SlurEvent` se deben añadir 'dentro' de la nota (o más concretamente, dentro de la expresión `EventChord`).

Ahora observamos la entrada:

```
(make-music
  'SequentialMusic
  'elements
  (list (make-music
          'EventChord
          'elements
          (list (make-music
                  'NoteEvent
                  'duration
                  (ly:make-duration 2 0 1 1)
                  'pitch
                  (ly:make-pitch 0 5 0))))))
```

Así pues, en nuestra función, tenemos que clonar esta expresión (de forma que tengamos dos notas para construir la secuencia), añadir `SlurEvents` a la propiedad `'elements` de cada una de ellas, y por último hacer una secuencia `SequentialMusic` con los dos `EventChords`.

```
doubleSlur = #(define-music-function (parser location note) (ly:music?)
         "Return: { note ( note ) }.
```

```
        `note' is supposed to be an EventChord."
        (let ((note2 (ly:music-deep-copy note)))
          (set! (ly:music-property note 'elements)
                (cons (make-music 'SlurEvent 'span-direction -1)
                      (ly:music-property note 'elements)))
          (set! (ly:music-property note2 'elements)
                (cons (make-music 'SlurEvent 'span-direction 1)
                      (ly:music-property note2 'elements)))
          (make-music 'SequentialMusic 'elements (list note note2))))
```

6.3.4 Añadir articulación a las notas (ejemplo)

La manera fácil de añadir articulación a las notas es fundir dos expresiones musicales en un contexto único, como está explicado en Sección 5.1.2 [Crear contextos], página 390. Sin embargo, suponga que queremos escribir una función musical que haga esto.

Una `$variable` dentro de la notación `#{...#}` es como usar una `\variable` normal en la notación clásica de LilyPond. Sabemos que

```
{ \musica -. -> }
```

no funciona en LilyPond. Podemos evitar este problema adjuntando la articulación a una nota de mentira,

```
{ << \musica s1*0-.-> }
```

pero a los efectos de este ejemplo, aprenderemos ahora cómo hacerlo en Scheme. Comenzamos examinando nuestra entrada y la salida deseada:

```
%  entrada
\displayMusic c4
===>
(make-music
  'EventChord
  'elements
  (list (make-music
          'NoteEvent
          'duration
          (ly:make-duration 2 0 1 1)
          'pitch
          (ly:make-pitch -1 0 0))))
=====
%  salida deseada
\displayMusic c4->
===>
(make-music
  'EventChord
  'elements
  (list (make-music
          'NoteEvent
          'duration
          (ly:make-duration 2 0 1 1)
          'pitch
          (ly:make-pitch -1 0 0))
        (make-music
          'ArticulationEvent
          'articulation-type
```

```
        "marcato")))
```

Vemos que una nota (`c4`) se representa como una expresión `EventChord`, con una expresión `NoteEvent` en su lista de elementos. Para añadir una articulación marcato, se debe añadir una expresión `ArticulationEvent` a la propiedad elementos de la expresión `EventChord`.

Para construir esta función, empezamos con

```
(define (add-marcato event-chord)
  "Añadir una ArticulationEvent de marcato a los elementos de `event-chord',
  que se ssupone que es una expresión EventChord."
  (let ((result-event-chord (ly:music-deep-copy event-chord)))
    (set! (ly:music-property result-event-chord 'elements)
          (cons (make-music 'ArticulationEvent
                  'articulation-type "marcato")
                (ly:music-property result-event-chord 'elements)))
    result-event-chord))
```

La primera línea es la forma de definir una función en Scheme: el nombre de la función es `add-marcato`, y tiene una variable llamada `event-chord`. En Scheme, el tipo de variable suele quedar claro a partir de su nombre (¡esto también es una buena práctica en otros lenguajes de programación!).

```
"Añadir una ArticulationEvent de marcato..."
```

es una descripción de lo que hace la función. No es estrictamente necesario, pero como los nombres de variable claros, es una buena práctica.

```
(let ((result-event-chord (ly:music-deep-copy event-chord)))
```

`let` se usa para declarar variables locales. Aquí usamos una variable local, llamada `result-event-chord`, a la que le damos el valor `(ly:music-deep-copy event-chord)`. `ly:music-deep-copy` es una función específica de LilyPond, como todas las funciones que comienzan por `ly:`. Se usa para hacer una copia de una expresión musical. Aquí, copiamos `event-chord` (el parámetro de la función). Recuerde que el propósito es añadir un marcato a una expresión `EventChord`. Es mejor no modificar el `EventChord` que se dio como argumento, porque podría utilizarse en algún otro lugar.

Ahora tenemos un `result-event-chord`, que es una expresión `NoteEventChord` y es una copia de `event-chord`. Añadimos el marcato a su propiedad lista de elementos.

```
(set! place new-value)
```

Aquí, lo que queremos establecer (el 'place') es la propiedad 'elements' de la expresión `result-event-chord`.

```
(ly:music-property result-event-chord 'elements)
```

`ly:music-property` es la función que se usa para acceder a las propiedades musicales (los `'elements`, `'duration`, `'pitch`, etc., que vemos en la salida de `\displayMusic` más arriba). El nuevo valor es la anterior propiedad elements, con un elemento adicional: la expresión `ArticulationEvent`, que copiamos a partir de la salida de `\displayMusic`,

```
(cons (make-music 'ArticulationEvent
        'articulation-type "marcato")
      (ly:music-property result-event-chord 'elements))
```

`cons` se usa para añadir un elemento a una lsita sin modificar la lista original. Esto es lo que queremos: la misma lista que antes, más la nueva expresión `ArticulationEvent`. El orden dentro de la propiedad elements no es importante aquí.

Finalmente, una vez añadida la articulación marcato a su propiedad `elements`, podemos devolver `result-event-chord`, de aquí la última línea de la función.

Ahora transformamos la función `add-marcato` en una función musical,

```
addMarcato = #(define-music-function (parser location event-chord)
                                      (ly:music?)
    "Add a marcato ArticulationEvent to the elements of `event-chord',
    which is supposed to be an EventChord expression."
    (let ((result-event-chord (ly:music-deep-copy event-chord)))
      (set! (ly:music-property result-event-chord 'elements)
            (cons (make-music 'ArticulationEvent
                    'articulation-type "marcato")
                  (ly:music-property result-event-chord 'elements)))
      result-event-chord))
```

Podemos verificar que esta función musical funciona correctamente,

```
\displayMusic \addMarcato c4
```

6.4 Interfaz de marcado para el programador

Los marcados están implementados como funciones de Scheme especiales que producen un elemento Stencil (sello) dado un número de argumentos.

6.4.1 Construcción del marcado en Scheme

El macro `markup` construye expresiones de marcado en Scheme, proporcionando una sintaxis similar a la de LilyPond. Por ejemplo:

```
(markup #:column (#:line (#:bold #:italic "hola" #:raise 0.4 "mundo")
                  #:larger #:line ("fulano" "fulanito" "menganito")))
```

equivale a:

```
\markup \column { \line { \bold \italic "hola" \raise #0.4 "mundo" }
                  \larger \line { fulano fulanito menganito } }
```

Este ejemplo muestra las principales reglas de traducción entre la sitaxis del marcado normal de LilyPond y la sintaxis del marcado de Scheme.

LilyPond	**Scheme**
`\markup marcado1`	`(markup marcado1)`
`\markup { marcado1 marcado2 ... }`	`(markup marcado1 marcado2 ... )`
`\instruccion`	`#:instruccion`
`\variable`	`variable`
`\center-column { ... }`	`#:center-column ( ... )`
`cadena`	`"cadena"`
`#argumento-de-scheme`	`argumento-de-scheme`

Todo el lenguaje Scheme está accesible dentro del macro `markup`. Por ejemplo, podemos usar llamadas a funciones dentro de `markup` para así manipular cadenas de caracteres. Esto es útil si se están definiendo instrucciones de marcado nuevas (véase Sección 6.4.3 [Definición de una instrucción de marcado nueva], página 444).

Advertencias y problemas conocidos

El argumento markup-list de instrucciones como `#:line`, `#:center` y `#:column` no pueden se una variable o el resultado de la llamada a una función.

```
(markup #:line (funcion-que-devuelve-marcados))
```

no es válido. Hay que usar las funciones `make-line-markup`, `make-center-markup` o `make-column-markup` en su lugar:

```
(markup (make-line-markup (funcion-que-devuelve-marcados)))
```

6.4.2 Cómo funciona internamente el marcado

En un elemento de marcado como

```
\raise #0.5 "ejemplo de texto"
```

`\raise` se representa en realidad por medio de la función `raise-markup`. La expresión de marcado se almacena como

```
(list raise-markup 0.5 (list simple-markup "ejemplo de texto"))
```

Cuando el marcado se convierte en objetos imprimibles (Stencils o sellos), se llama la función `raise-markup` como

```
(apply raise-markup
       \objeto de marcado
       lista de listas asociativas de propiedades
       0.5
       el marcado "ejemplo de texto")
```

Primero la función `raise-markup` crea el sello para la cadena `ejemplo de texto`, y después eleva el sello Stencil en 0.5 espacios de pentagrama. Este es un ejemplo bastante simple; en el resto de la sección podrán verse ejemplos más complejos, así como en '`scm/define-markup-commands.scm`'.

6.4.3 Definición de una instrucción de marcado nueva

Las instrucciones de marcado nuevas se pueden definir con el macro de Scheme `define-markup-command`.

```
(define-markup-command (nombre-de-la-instruccion layout props arg1 arg2 ...)
            (arg1-type? arg2-type? ...)
  ..command body..)
```

Los argumentos son

argi *i*-ésimo argumento de la instrucción

argi-type? predicado de tipo para el argumento *i*-ésimo

layout la definición de 'presentación'

props lista de listas asociativas, que contiene todas las propiedades activas.

Como ejemplo sencillo, mostramos cómo añadir una instrucción `\smallcaps`, que selecciona una tipografía de versalitas. Normalmente podríamos seleccionar la tipografía de versalitas,

```
\markup { \override #'(font-shape . caps) Texto-en-versalitas }
```

Esto selecciona la tipografía de versalitas mediante el establecimiento de la propiedad `font-shape` a `#'caps` para la interpretación de `Texto-en-versalitas`.

Para poner lo anterior disponible como la instrucción `\smallcaps`, tenemos que definir una función utilizando `define-markup-command`. La instrucción ha de tomar un argumento del tipo `markup`. Por tanto, el inicio de la definición ha de ser

```
(define-markup-command (smallcaps layout props argument) (markup?)
```

Lo que aparece a continuación es el contenido de la instrucción: debemos interpretar el `argument` como un marcado, es decir:

```
(interpret-markup layout ... argument)
```

Esta interpretación tiene que añadir `'(font-shape . caps)` a las propiedades activas, por lo que sustituimos lo siguiente por los ... en el ejemplo anterior:

```
(cons (list '(font-shape . caps) ) props)
```

La variable `props` es una lista de a-listas, y se lo anteponemos haciendo la operación cons de una lista con el ajuste adicional.
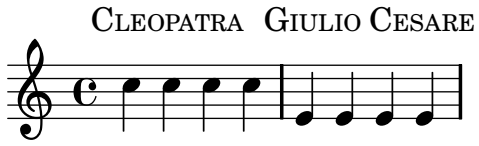
Supongamos que estamos tipografiando un recitativo de una ópera y nos gustaría definir una instrucción que presente los nombres de los personajes de una forma personalizada. Queremos que los nombres se impriman con versalitas y se desplacen un poco a la izquierda y hacia arriba. Definimos una instrucción `\character` que toma en cuenta la traslación necesaria y utiliza la instrucción `\smallcaps` recién definida:

```
#(define-markup-command (character layout props nombre) (string?)
  "Imprimir el nombre del personaje en versalitas, desplazado a la izquierda y hacia
  arriba.  Sintaxis: \\character #\"nombre\""
  (interpret-markup layout props
   (markup #:hspace 0 #:translate (cons -3 1) #:smallcaps nombre)))
```

Esta es una complicación que requiere una explicación: los textos por encima y por debajo del pentagrama se mueven verticalmente de forma que estén a una cierta distancia (la propiedad `padding`) del pentagrama y de las notas. Para asegurar que este mecanismo no anula el efecto de nuestro `#:translate`, añadimos una cadena vacía (`#:hspace 0`) antes del texto trasladado. Ahora el `#:hspace 0` se pone encima de las notas, y el `nombre` se mueve en relación a dicha cadena vacía. El efecto neto es que el texto se mueve hacia la izquierda y hacia arriba.

El resultado final es como sigue:

```
{
  c''^\markup \character #"Cleopatra"
  e'^\markup \character #"Giulio Cesare"
}
```



Hemos usado la forma de fuente tipográfica `caps`, pero supongamos que nuestra fuente no tiene la variante de versalitas. En ese caso tenemos que hacer una falsa fuente de mayúsculas pequeñas haciendo que la cadena en mayúsculas tenga la primera legra un poco mayor:

```
#(define-markup-command (smallcaps layout props str) (string?)
  "Print the string argument in small caps."
  (interpret-markup layout props
   (make-line-markup
    (map (lambda (s)
          (if (= (string-length s) 0)
              s
              (markup #:large (string-upcase (substring s 0 1))
                      #:translate (cons -0.6 0)
                      #:tiny (string-upcase (substring s 1)))))
         (string-split str #\Space)))))
```

La instrucción `smallcaps` primero divide su argumento de cadena en unidades o palabras separadas por espacios (`(string-split str #\Space)`); para cada unidad o palabra, se construye un marcado con la primera letra agrandada y en mayúscula (`#:large (string-upcase (substring s 0 1))`), y un segundo marcado construido con las letras siguientes reducidas de tamaño y en mayúsculas (`#:tiny (string-upcase (substring s 1))`). Como LilyPond introduce un espacio entre los marcados de una misma línea, el segundo marcado se traslada a

la izquierda (`#:translate (cons -0.6 0) ...`). Después, los marcados construidos para cada palabra se ponen en una línea mediante `(make-line-markup ...)`. Finalmente, el marcado resultante se pasa a la función `interpret-markup`, con los argumentos `layout` y `props`.

Nota: ahora existe una instrucción interna `\smallCaps` que se puede usar para poner texto en versalitas. Consulte Sección B.8 [Text markup commands], página 476, para ver más detalles.

Advertencias y problemas conocidos

Actualmente las combinaciones de argumentos que hay disponibles (después de los argumentos estándar *layout* y *props*) para una instrucción de marcado definida con `define-markup-command` se limitan a la siguiente lista:

(ningún argumento)
list
markup
markup markup
scm
scm markup
scm scm
scm scm markup
scm scm markup markup
scm markup markup
scm scm scm

En la tabla de arriba, *scm* representa los tipos de datos nativos de Scheme como 'number' (número) o 'string' (cadena).

Como ejemplo, no es posible usar una instrucción de marcado `fulanito` con cuatro argumentos definida como

```
#(define-markup-command (fulanito layout props
                          num1    str1    num2    str2)
                         (number? string? number? string?)
  ...)
```

Si la aplicamos como, digamos,

```
\markup \fulanito #1 #"mengano" #2 #"zutano"
```

`lilypond` protesta diciendo que no puede analizar `fulanito` debido a su firma de Scheme desconocida.

6.4.4 Definición de nuevas instrucciones de lista de marcado

Las instrucciones de listas de marcado se definen con el macro de Scheme `define-markup-list-command`, que es similar al macro `define-markup-command` descrito en Sección 6.4.3 [Definición de una instrucción de marcado nueva], página 444, excepto que donde éste devuelve un sello único, aquél devuelve una lista de sellos.

En el siguiente ejemplo se define una instrucción de lista de marcado `\paragraph`, que devuelve una lista de líneas justificadas, estando la primera de ellas sangrada. La anchura del sangrado se toma del argumento `props`.

```
#(define-markup-list-command (paragraph layout props args) (markup-list?)
   (let ((indent (chain-assoc-get 'par-indent props 2)))
     (interpret-markup-list layout props
       (make-justified-lines-markup-list (cons (make-hspace-markup indent)
                                               args)))))
```

Aparte de los argumentos usuales `layout` y `props`, la instrucción de lista de marcados `paragraph` toma un argumento de lista de marcados, llamado `args`. El predicado para listas de marcados es `markup-list?`.

Em primer lugar, la función toma el ancho del sangrado, una propiedad llamada aquí `par-indent`, de la lista de propiedades `props`. Si no se encuentra la propiedad, el valor predeterminado es 2. Después, se hace una lista de líneas justificadas usando la función `make-justified-lines-markup-list`, que está relacionada con la instrucción incorporada de lista de marcados `\justified-lines`. Se añade un espacio horizontal al principio usando la función `make-hspace-markup`. Finalmente, la lista de marcados se interpreta usando la función `interpret-markup-list`.

Esta nueva instrucción de lista de marcados se puede usar como sigue:

```
\markuplines {
  \paragraph {
    El arte de la tipografía musical se llama  \italic {grabado (en plancha).}
    El término deriva del proceso tradicional de impresión de música.
    hace sólo algunas décadas, las partituras se hacían cortando y estampando
    la música en una plancha de zinc o lata en una imagen invertida.
  }
  \override-lines #'(par-indent . 4) \paragraph {
    La plancha se tenía que entintar, y las depresiones causadas por los cortes
    y estampados retienen la tinta.  Se formaba una imagen presionando el papel
    contra la plancha.  El estampado y cortado se hacía completamente
    a mano.
  }
}
```

6.5 Contextos para programadores

6.5.1 Evaluación de contextos

Se pueden modificar los contextos durante la interpretación con código de Scheme. La sintaxis para esto es

`\applyContext` *función*

función debe ser una función de Scheme que toma un único argumento, que es el contexto al que aplicarla. El código siguiente imprime el número del compás actual sobre la salida estándar durante la compilación:

```
\applyContext
  #(lambda (x)
    (format #t "\nSe nos ha llamado en el compás número ~a.\n"
     (ly:context-property x 'currentBarNumber)))
```

6.5.2 Ejecutar una función sobre todos los objetos de la presentación

La manera más versátil de realizar el ajuste fino de un objeto es `\applyOutput`. Su sintaxis es

`\applyOutput` *contexto proc*

donde *proc* es una función de Scheme, que toma tres argumentos.

Al interpretarse, la función *proc* se llama para cada objeto de presentación que se encuentra en el contexto *contexto*, con los siguientes argumentos:

- el propio objeto de presentación,
- el contexto en que se creó el objeto de presentación, y
- el contexto en que se procesa `\applyOutput`.

Además, la causa del objeto de presentación, es decir el objeto o expresión musical que es responsable de haberlo creado, está en la propiedad `cause` del objeto. Por ejemplo, para la

cabeza de una nota, éste es un evento Sección “NoteHead” in *Referencia de Funcionamiento Interno*, y para un objeto Sección “Stem” in *Referencia de Funcionamiento Interno* (plica), éste es un objeto Sección “Stem” in *Referencia de Funcionamiento Interno*.

He aquí una función que usar para `\applyOutput`; borra las cabezas de las notas que están sobre la línea central:

```
(define (blanker grob grob-origin context)
 (if (and (memq (ly:grob-property grob 'interfaces)
                note-head-interface)
          (eq? (ly:grob-property grob 'staff-position) 0))
     (set! (ly:grob-property grob 'transparent) #t)))
```
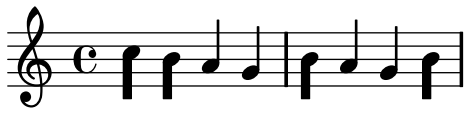
6.6 Procedimientos de Scheme como propiedades

Las propiedades (como el grosor, la dirección, etc.) se pueden establecer a valores fijos con \override, p. ej.

```
\override Stem #'thickness = #2.0
```

Las propiedades pueden fijarse también a un procedimiento de scheme,

```
\override Stem #'thickness = #(lambda (grob)
    (if (= UP (ly:grob-property grob 'direction))
        2.0
        7.0))
c b a g b a g b
```



En este caso, el procedimiento se ejecuta tan pronto como el valor de la propiedad se reclama durante el proceso de formateo.

Casi todo el motor de tipografiado está manejado por estos *callbacks*. Entre las propiedades que usan normalmente *callbacks* están

`stencil` La rutina de impresión, que construye un dibujo para el símbolo

`X-offset` La rutina que establece la posición horizontal

`X-extent` La rutina que calcula la anchura de un objeto

El procedimiento siempre toma un argumento único, que es el grob (el objeto gráfico).

Si se deben llamar rutinas con varios argumentos, el grob actual se puede insertar con una cerradura de grob. He aquí un ajuste procedente de `AccidentalSuggestion`,

```
(X-offset .
  ,(ly:make-simple-closure
    `(,+
        ,(ly:make-simple-closure
           (list ly:self-alignment-interface::centered-on-x-parent))
      ,(ly:make-simple-closure
           (list ly:self-alignment-interface::x-aligned-on-self)))))
```

En este ejemplo, tanto `ly:self-alignment-interface::x-aligned-on-self` como `ly:self-alignment-interface::centered-on-x-parent` se llaman con el grob como argumento. El resultado se añade con la función `+`. Para asegurar que esta adición se ejecuta adecuadamente, todo ello se encierra dentro de `ly:make-simple-closure`.

De hecho, usar un solo procedimiento como valor de una propiedad equivale a

```
(ly:make-simple-closure (ly:make-simple-closure (list proc)))
```

El `ly:make-simple-closure` interior aporta el grob como argumento de *proc*, el exterior asegura que el resultado de la función es lo que se devuelve, en lugar del objeto `simple-closure`.

6.7 Usar código de Scheme en lugar de \tweak

La principal desventaja de `\tweak` es su inflexibilidad sintáctica. Por ejemplo, lo siguiente produce un error de sintaxis.

```
F = \tweak #'font-size #-3 -\flageolet

\relative c'' {
  c4^\F c4_\F
}
```

En otras palabras, `\tweak` no se comporta como una articulación en cuando a la sintaxis; concretamente, no se puede adjuntar con `^` y `_`.

Usando Scheme, se puede dar un rodeo a este problema. La ruta hacia el resultado se da en Sección 6.3.4 [Añadir articulación a las notas (ejemplo)], página 441, especialmente cómo usar `\displayMusic` como guía de ayuda.

```
F = #(let ((m (make-music 'ArticulationEvent
                          'articulation-type "flageolet")))
       (set! (ly:music-property m 'tweaks)
             (acons 'font-size -3
                    (ly:music-property m 'tweaks)))
       m)

\relative c'' {
  c4^\F c4_\F
}
```

Aquí, las propiedades `tweaks` del objeto flageolet `m` (creado con `make-music`) se extraen con `ly:music-property`, se antepone un nuevo par clave-valor para cambiar el tamaño de la tipografía a la lista de propiedades con la función de Scheme `acons`, y finalmente el resultado se escribe de nuevo con `set!`. El último elemento del bloque `let` es el valor de retorno, el propio `m`.

6.8 Trucos difíciles

Hay un cierto número de tipos de ajustes difíciles.

- Un tipo de ajuste difícil es la apariencia de los objetos de extensión, como las ligaduras de expresión y de unión. Inicialmente, sólo se crea uno de estos objetos, y pueden ajustarse con el mecanismo normal. Sin embargo, en ciertos casos los objetos extensores cruzan los saltos de línea. Si esto ocurre, estos objetos se clonan. Se crea un objeto distinto por cada sistema en que se encuentra. Éstos son clones del objeto original y heredan todas sus propiedades, incluidos los `\override`s.

  En otras palabras, un `\override` siempre afecta a todas las piezas de un objeto de extensión fragmentado. Para cambiar sólo una parte de un extensor en el salto de línea, es necesario inmiscuirse en el proceso de formateado. El *callback* `after-line-breaking` contiene el procedimiento Scheme que se llama después de que se han determinado los saltos de línea, y los objetos de presentación han sido divididos sobre los distintos sistemas.

  En el ejemplo siguiente, definimos un procedimiento `my-callback`. Este procedimiento

  - determina si hemos sido divididos por los saltos de línea
  - en caso afirmativo, reúne todos los objetos divididos

- comprueba si somos el último de los objetos divididos
- en caso afirmativo, establece `extra-offset`.

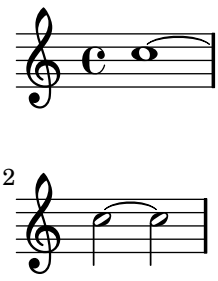
Este procedimiento se instala en Sección "Tie" in *Referencia de Funcionamiento Interno* (ligadura de unión), de forma que la última parte de la ligadura dividida se traslada hacia arriba.

```
#(define (my-callback grob)
  (let* (
         ; have we been split?
         (orig (ly:grob-original grob))

         ; if yes, get the split pieces (our siblings)
         (siblings (if (ly:grob? orig)
                       (ly:spanner-broken-into orig) '() )))

   (if (and (>= (length siblings) 2)
            (eq? (car (last-pair siblings)) grob))
     (ly:grob-set-property! grob 'extra-offset '(-2 . 5)))))

\relative c'' {
  \override Tie #'after-line-breaking =
  #my-callback
  c1 ~ \break c2 ~ c
}
```



Al aplicar este truco, la nueva función de callback `after-line-breaking` también debe llamar a la antigua `after-line-breaking`, si existe. Por ejemplo, si se usa con `Hairpin`, se debe llamar también a `ly:hairpin::after-line-breaking`.

- Algunos objetos no se pueden cambiar con `\override` por razones técnicas. Son ejemplos `NonMusicalPaperColumn` y `PaperColumn`. Se pueden cambiar con la función `\overrideProperty` que funciona de forma similar a `\once \override`, pero usa una sintaxis distinta.

```
\overrideProperty
#"Score.NonMusicalPaperColumn"  % Nombre del grob
#'line-break-system-details     % Nombre de la propiedad
#'((next-padding . 20))         % Valor
```

Observe, sin embargo, que `\override`, aplicado a `NoteMusicalPaperColumn` y a `PaperColumn`, aún funciona como se espera dentro de los bloques `\context`.

Apéndice A Lista bibliográfica

Si tiene necesidad de aprender más acerca de la notación musical, le presentamos a continuación algunos títulos interesantes que puede leer.

Ignatzek 1995

Klaus Ignatzek, Die Jazzmethode für Klavier. Schott's Söhne 1995. Mainz, Germany ISBN 3-7957-5140-3.

Instructiva introducción a la interpretación de Jazz al piano. Uno de los primeros capítulos contiene una panorámica de los acordes más comunes de la música de Jazz.

Gerou 1996

Tom Gerou and Linda Lusk, Essential Dictionary of Music Notation. Alfred Publishing, Van Nuys CA ISBN 0-88284-768-6.

Una lista concisa y ordenada alfabéticamente de los problemas de la composición tipográfica y la notación musical, que abarca la mayor parte de los casos más comunes.

Read 1968

Gardner Read, Music Notation: A Manual of Modern Practice. Taplinger Publishing, New York (2nd edition).

Una obra estándar sobre notación musical.

Ross 1987 Ted Ross, Teach yourself the art of music engraving and processing. Hansen House, Miami, Florida 1987.

Este libro trata del grabado musical, es decir, composición tipográfica profesional. Contiene instrucciones sobre el estampado, la utilización de las plumillas y las convenciones notacionales. También son interesantes las secciones sobre los tecnicismos y la historia de la reproducción.

Schirmer 2001

The G.Schirmer/AMP Manual of Style and Usage. G.Schirmer/AMP, NY, 2001. (Este libro se puede pedir al departamento de alquiler.)

Este manual se centra específicamente en la preparación de los manuscritos para la publicación por Schirmer. Discute muchos detalles que no se pueden encontrar en otros libros de notación más normales. También proporciona una buena idea sobre lo que se necesita para llevar la impresión hasta la calidad editorial.

Stone 1980

Kurt Stone, Music Notation in the Twentieth Century. Norton, New York 1980.
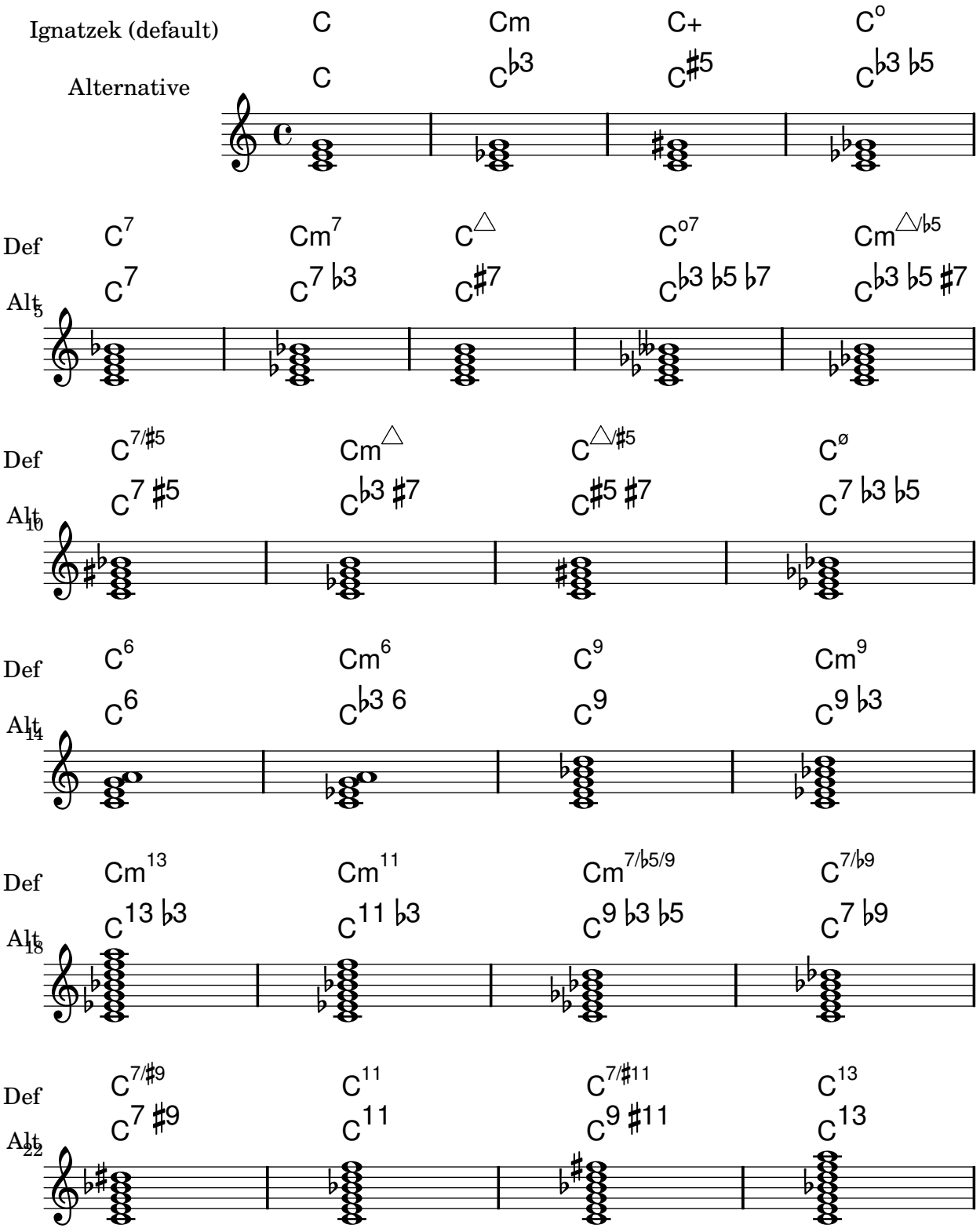
Este libro describe la notación musical para la música seria moderna, pero empieza por una amplia panorámica de las prácticas existentes de la notación tradicional.

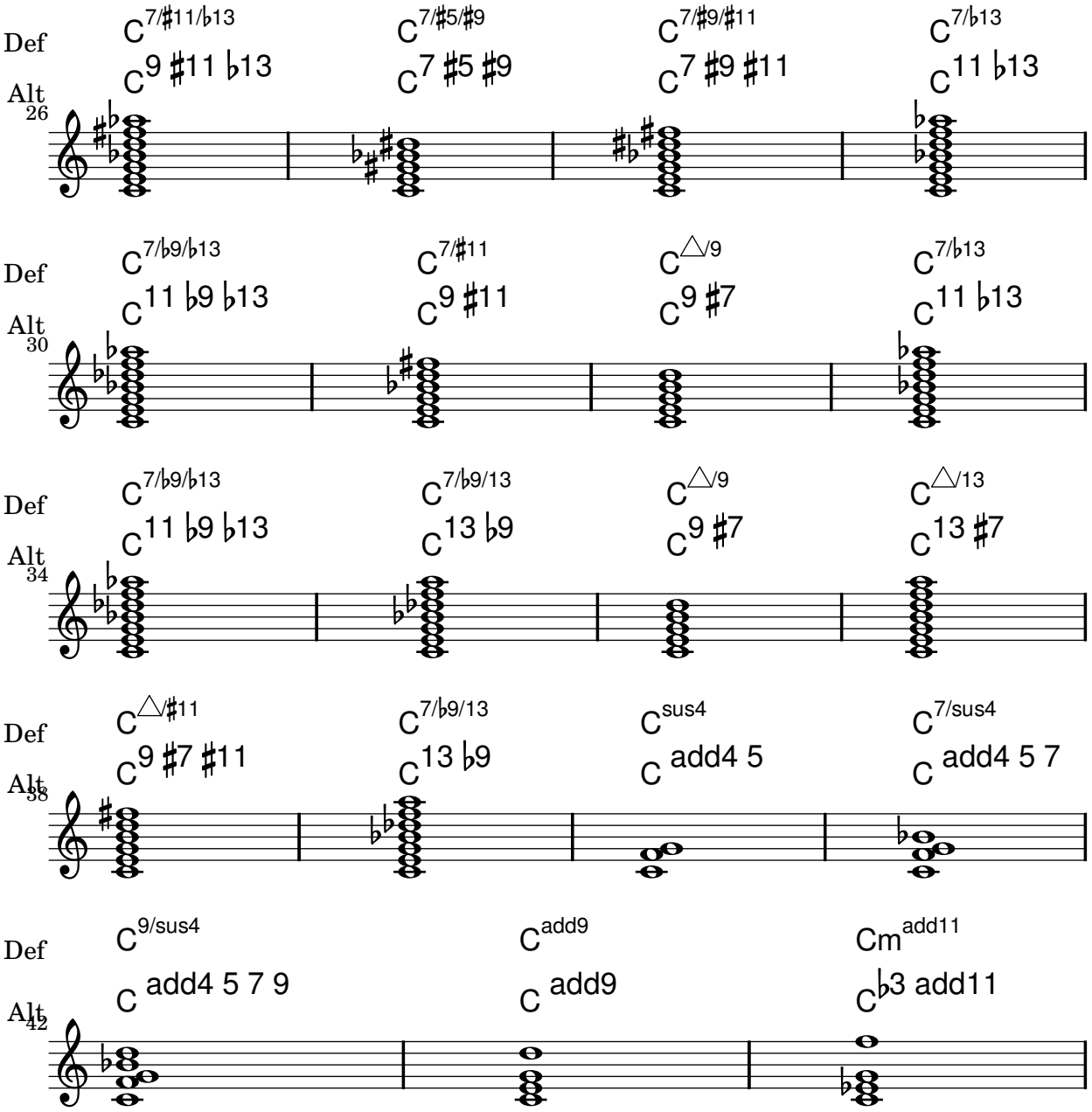
El archivo de código fuente incluye una bibliografía de BibTEX más elaborada con más de 100 elementos, en '`Documentation/bibliography/`'.

Apéndice B Tablas del manual sobre notación

B.1 Carta de nombres de acordes

Las siguientes tablas muestran dos sistemas estándar para imprimir los nombres de los acordes, junto a las notas que representan.

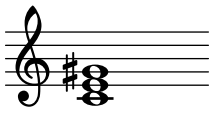
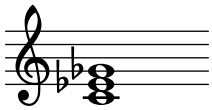
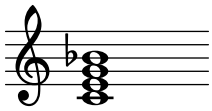
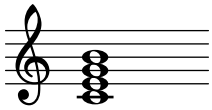
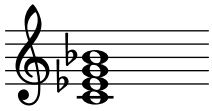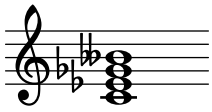
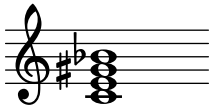
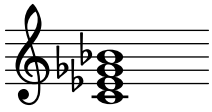
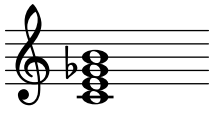


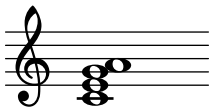
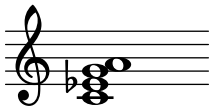
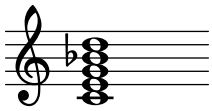
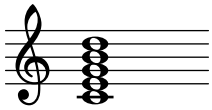
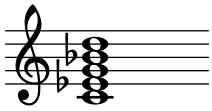
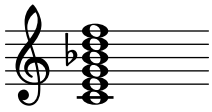
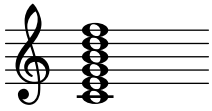
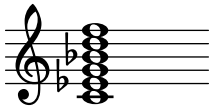
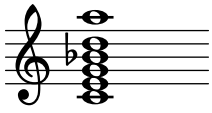


B.2 Modificadores de acorde más usuales

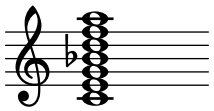
La tabla siguiente presenta los modificadores de acorde que se pueden usar en el modo de acordes `\chordmode` para generar las estructuras de acordes estándar.

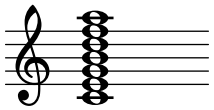
Tipo de acorde	**Intervalos**	**Modificador(es)**	**Ejemplo**
Mayor	Tercera mayor y quinta justa	`5` o nada	
Menor	Tercera menor y quinta justa	`m` o `m5`	

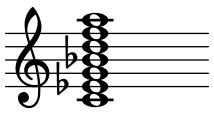
Aumentado	Tercera mayor y quinta aumentada	`aug`	
Disminuido	Tercera menor y quinta disminuida	`dim`	
Séptima dominante	Tríada mayor con séptima menor	`7`	
Séptima mayor	Tríada mayor y séptima mayor	`maj7` or `maj`	
Séptima menor	Tríada menor y séptima menor	`m7`	
Séptima disminuida	Tríada disminuida y séptima disminuida	`dim7`	
Aumentado con séptima	Tríada aumentada y séptima menor	`aug7`	
Séptima semi-disminuido	Tríada disminuida y séptima menor	`m7.5-`	
Menor con séptima mayor	Tríada menor con séptima mayor	`maj7.5-`	

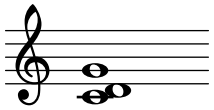
Mayor con sexta	Tríada mayor y sexta	`6`	
Menor con sexta	Tríada menor y sexta	`m6`	
Novena dominante	Séptima dominante con novena mayor	`9`	
Mayor con novena	Séptima mayor y novena mayor	`maj9`	
Menor con novena	Séptima menor y novena mayor	`m9`	
Oncena dominante	Novena dominante con oncena justa	`11`	
Oncena mayor	Novena mayor y oncena justa	`maj11`	
Menor con oncena	Novena menor y oncena justa	`m11`	
Trecena dominante	Novena dominante y trecena mayor	`13`	

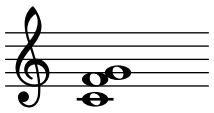
Trecena dominante	Oncena dominante y trecena mayor	`13.11`
Mayor con trecena	Oncena mayor y trecena mayor	`maj13.11`
Menor con trecena	Oncena menor y trecena mayor	`m13.11`
Segunda suspendida	Segunda mayor y quinta justa	`sus2`
Cuarta suspendida	Cuarta justa y quinta justa	`sus4`





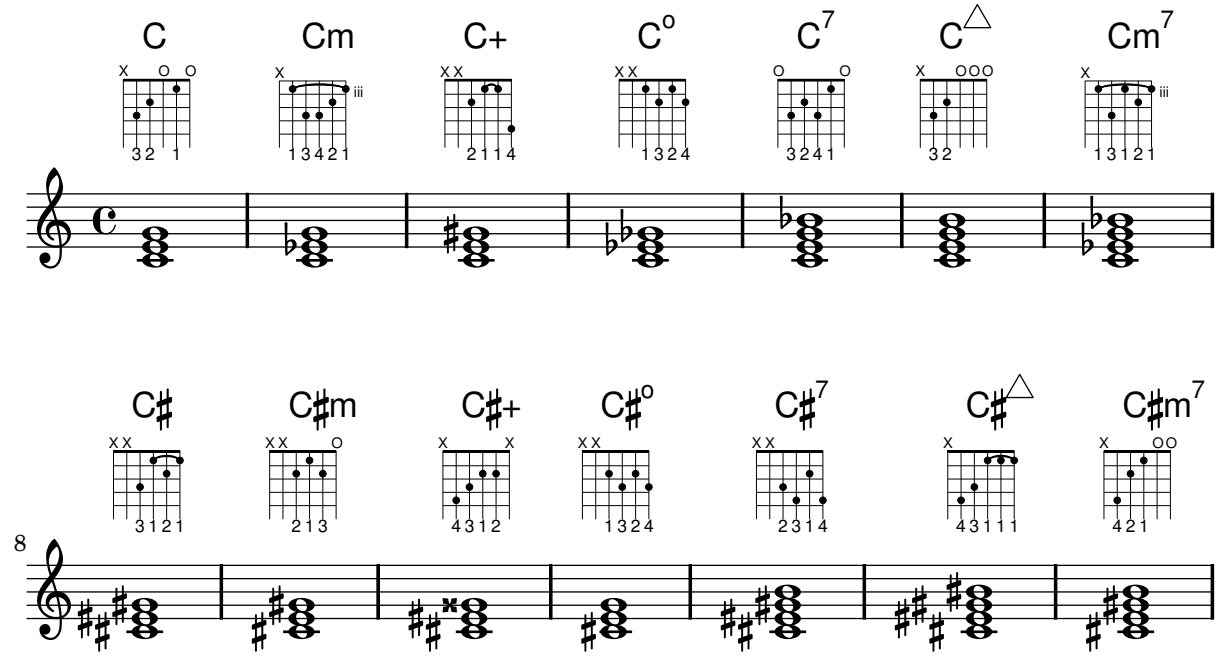


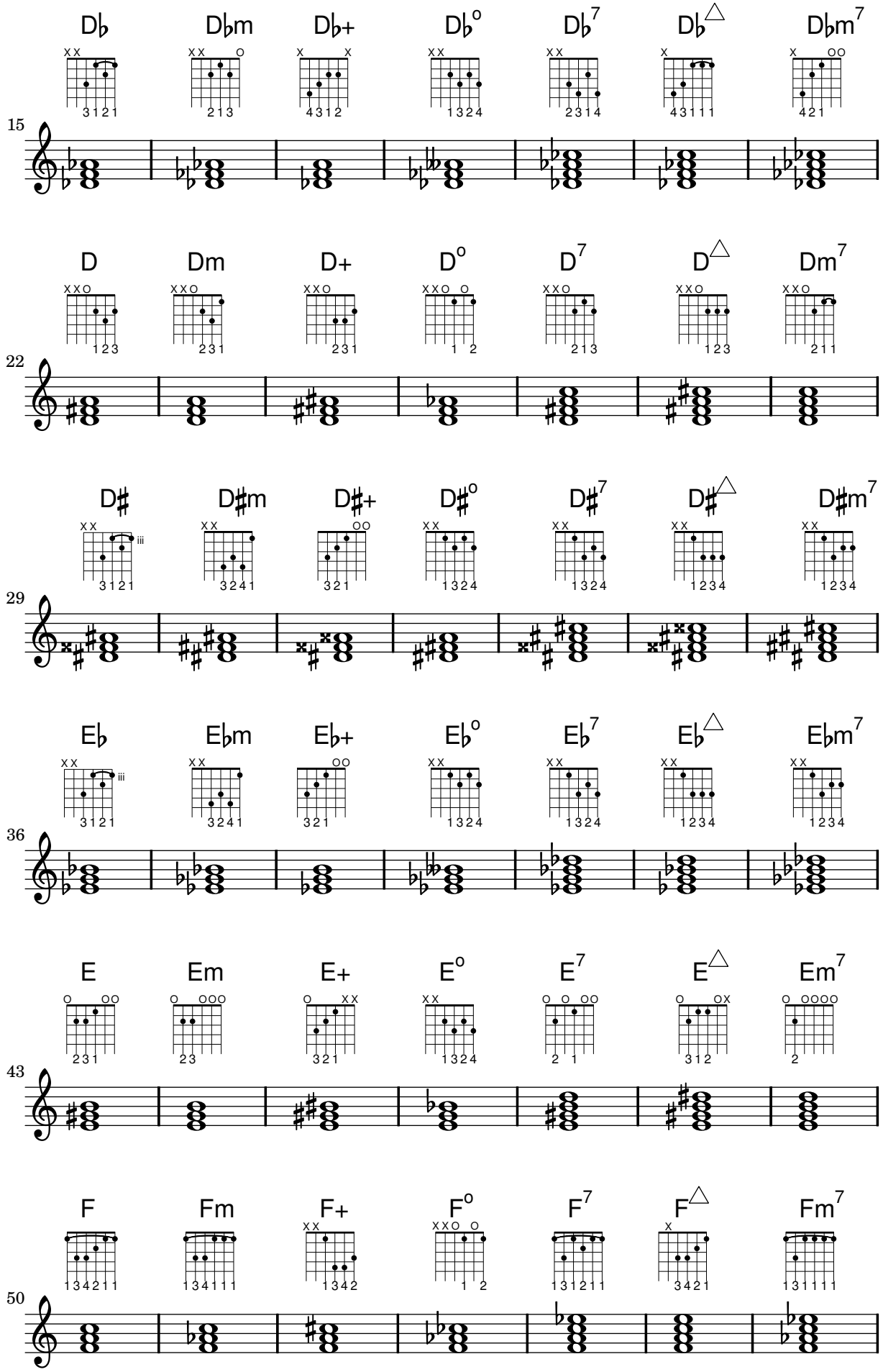


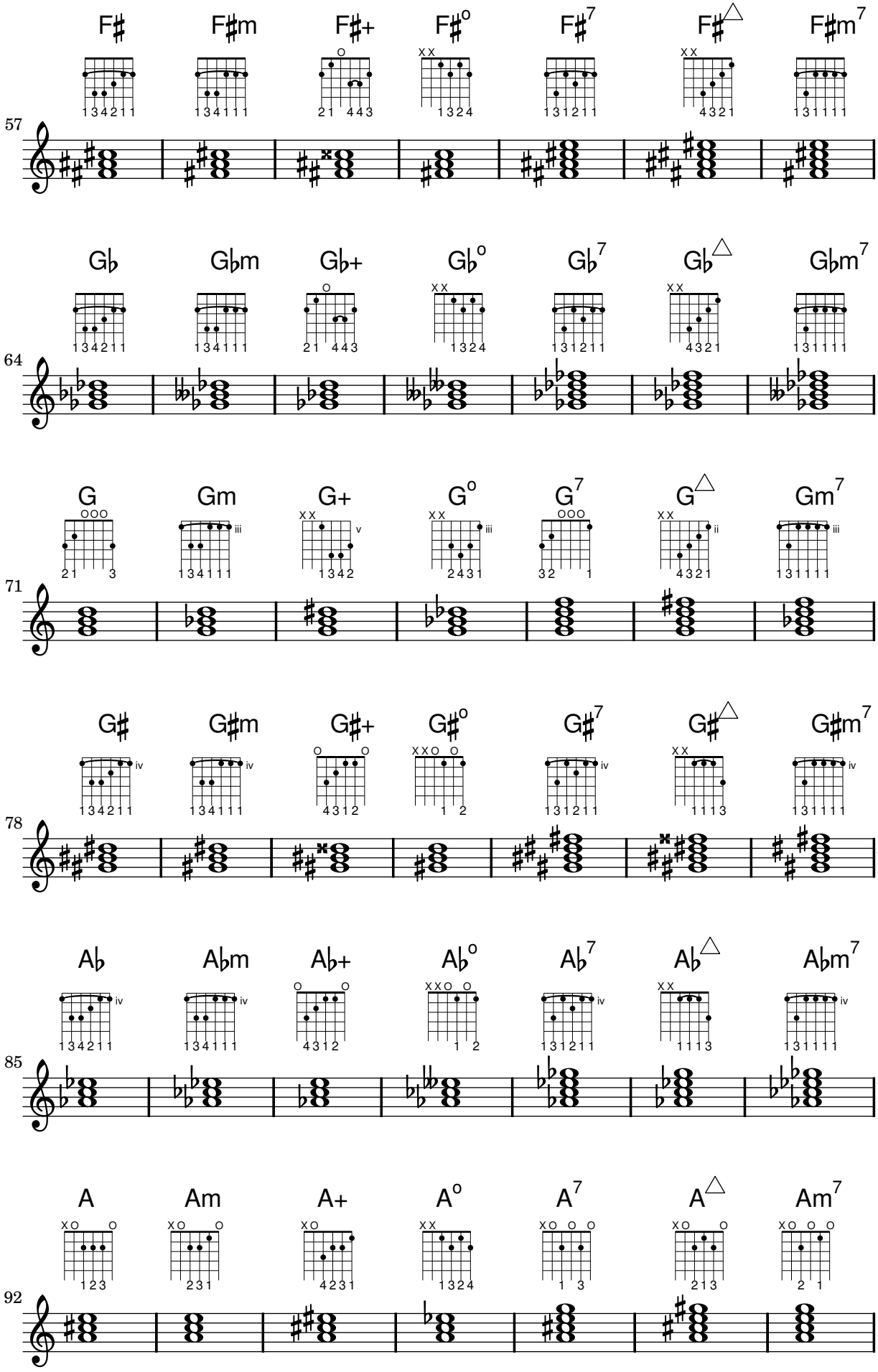


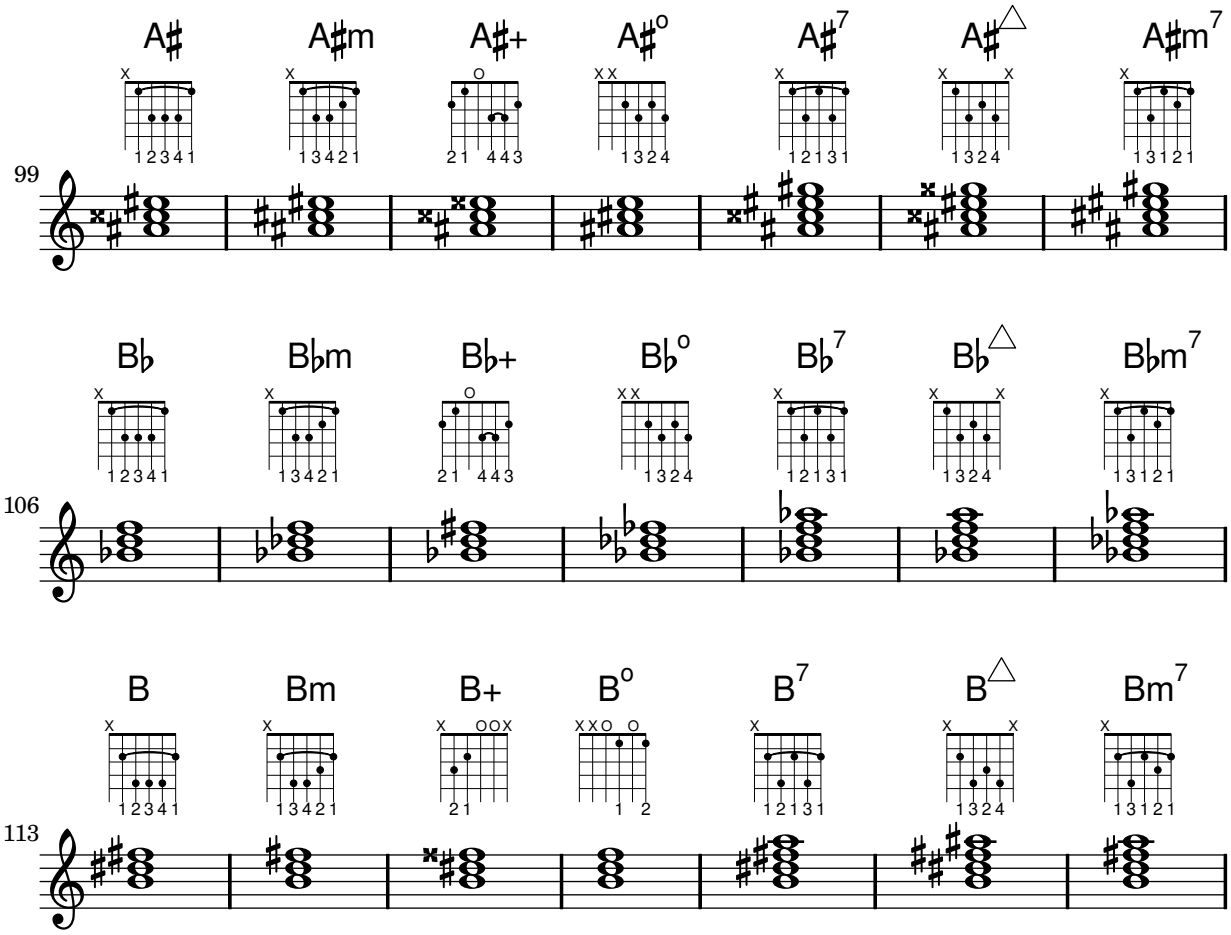
B.3 Diagramas predefinidos de trastes

El cuadro siguiente presenta los diagramas de traste predefinidos.




D♭ D♭m D♭+ D♭° D♭7 D♭△ D♭m7
15
D Dm D+ D° D7 D△ Dm7
22
D♯ D♯m D♯+ D♯° D♯7 D♯△ D♯m7
29
E♭ E♭m E♭+ E♭° E♭7 E♭△ E♭m7
36
E Em E+ E° E7 E△ Em7
43
F Fm F+ F° F7 F△ Fm7
50


F♯ F♯m F♯+ F♯° F♯7 F♯△ F♯m7
57
G♭ G♭m G♭+ G♭° G♭7 G♭△ G♭m7
64
G Gm G+ G° G7 G△ Gm7
71
G♯ G♯m G♯+ G♯° G♯7 G♯△ G♯m7
78
A♭ A♭m A♭+ A♭° A♭7 A♭△ A♭m7
85
A Am A+ A° A7 A△ Am7
92



B.4 Instrumentos MIDI

A continuación figura una lista con los nombres que se pueden utilizar para la propiedad `midiInstrument` (instrumento MIDI).

```
acoustic grand
bright acoustic
electric grand
honky-tonk
electric piano 1
electric piano 2
harpsichord
clav
celesta
glockenspiel
music box
vibraphone
marimba
xylophone
tubular bells
dulcimer
drawbar organ
percussive organ
rock organ
church organ
reed organ
accordion
harmonica
contrabass
tremolo strings
pizzicato strings
orchestral strings
timpani
string ensemble 1
string ensemble 2
synthstrings 1
synthstrings 2
choir aahs
voice oohs
synth voice
orchestra hit
trumpet
trombone
tuba
muted trumpet
french horn
brass section
synthbrass 1
synthbrass 2
soprano sax
alto sax
lead 7 (fifths)
lead 8 (bass+lead)
pad 1 (new age)
pad 2 (warm)
pad 3 (polysynth)
pad 4 (choir)
pad 5 (bowed)
pad 6 (metallic)
pad 7 (halo)
pad 8 (sweep)
fx 1 (rain)
fx 2 (soundtrack)
fx 3 (crystal)
fx 4 (atmosphere)
fx 5 (brightness)
fx 6 (goblins)
fx 7 (echoes)
fx 8 (sci-fi)
sitar
banjo
shamisen
koto
kalimba
```

```
concertina                tenor sax           bagpipe
acoustic guitar (nylon)   baritone sax        fiddle
acoustic guitar (steel)   oboe                shanai
electric guitar (jazz)    english horn        tinkle bell
electric guitar (clean)   bassoon             agogo
electric guitar (muted)   clarinet            steel drums
overdriven guitar         piccolo             woodblock
distorted guitar          flute               taiko drum
guitar harmonics          recorder            melodic tom
acoustic bass             pan flute           synth drum
electric bass (finger)    blown bottle        reverse cymbal
electric bass (pick)      shakuhachi          guitar fret noise
fretless bass             whistle             breath noise
slap bass 1               ocarina             seashore
slap bass 2               lead 1 (square)     bird tweet
synth bass 1              lead 2 (sawtooth)   telephone ring
synth bass 2              lead 3 (calliope)   helicopter
violin                    lead 4 (chiff)      applause
viola                     lead 5 (charang)    gunshot
cello                     lead 6 (voice)
```

B.5 Lista de colores

Colores normales

La sintaxis para el uso de colores se encuentra detallada en [Colorear objetos], página 158.

```
black       white          red         green
blue        cyan           magenta     yellow
grey        darkred        darkgreen   darkblue
darkcyan    darkmagenta    darkyellow
```

Nombres de los colores de X

Los nombres de los colores de X admiten algunas variantes:

Un nombre que se escribe como una palabra única con mayúsculas intercaladas (p.ej. 'LightSlateBlue') se puede escribir también como palabras separadas por espacios y sin mayúsculas (p.ej. 'light slate blue').

La palabra 'grey' siempre se puede escribir como 'gray' (p.ej. 'DarkSlateGray').

Algunos nombres admiten un sufijo numérico (p.ej. 'LightSalmon4').

Nombres de los colores sin sufijo numérico:

```
snow             GhostWhite      WhiteSmoke      gainsboro       FloralWhite
OldLace          linen           AntiqueWhite    PapayaWhip      BlanchedAlmond
bisque           PeachPuff       NavajoWhite     moccasin        cornsilk
ivory            LemonChiffon    seashell        honeydew        MintCream
azure            AliceBlue       lavender        LavenderBlush   MistyRose
white            black           DarkSlateGrey   DimGrey         SlateGrey
LightSlateGrey   grey            LightGrey       MidnightBlue    navy
NavyBlue         CornflowerBlue  DarkSlateBlue   SlateBlue       MediumSlateBlue
LightSlateBlue   MediumBlue      RoyalBlue       blue            DodgerBlue
DeepSkyBlue      SkyBlue         LightSkyBlue    SteelBlue       LightSteelBlue
LightBlue        PowderBlue      PaleTurquoise   DarkTurquoise   MediumTurquoise
```

```
turquoise       cyan            LightCyan       CadetBlue       MediumAquamarine
aquamarine      DarkGreen       DarkOliveGreen  DarkSeaGreen    SeaGreen
MediumSeaGreen  LightSeaGreen   PaleGreen       SpringGreen     LawnGreen
green           chartreuse      MediumSpringGreen       GreenYellow     LimeGreen
YellowGreen     ForestGreen     OliveDrab       DarkKhaki       khaki
PaleGoldenrod   LightGoldenrodYellow    LightYellow     yellow  gold
LightGoldenrod  goldenrod       DarkGoldenrod   RosyBrown       IndianRed
SaddleBrown     sienna          peru            burlywood       beige
wheat           SandyBrown      tan             chocolate       firebrick
brown           DarkSalmon      salmon          LightSalmon     orange
DarkOrange      coral           LightCoral      tomato          OrangeRed
red             HotPink         DeepPink        pink            LightPink
PaleVioletRed   maroon          MediumVioletRed VioletRed       magenta
violet          plum            orchid          MediumOrchid    DarkOrchid
DarkViolet      BlueViolet      purple          MediumPurple    thistle
DarkGrey        DarkBlue        DarkCyan        DarkMagenta     DarkRed
LightGreen
```

Nombres de los colores con sufijo numérico

En los nombres siguientes, el sufijo N puede ser un número del 1 al 4:

```
snowN           seashellN       AntiqueWhiteN   bisqueN         PeachPuffN
NavajoWhiteN    LemonChiffonN   cornsilkN       ivoryN          honeydewN
LavenderBlushN  MistyRoseN      azureN          SlateBlueN      RoyalBlueN
blueN           DodgerBlueN     SteelBlueN      DeepSkyBlueN    SkyBlueN
LightSkyBlueN   LightSteelBlueN LightBlueN      LightCyanN      PaleTurquoiseN
CadetBlueN      turquoiseN      cyanN           aquamarineN     DarkSeaGreenN
SeaGreenN       PaleGreenN      SpringGreenN    greenN          chartreuseN
OliveDrabN      DarkOliveGreenN khakiN          LightGoldenrodN LightYellowN
yellowN         goldN           goldenrodN      DarkGoldenrodN  RosyBrownN
IndianRedN      siennaN         burlywoodN      wheatN          tanN
chocolateN      firebrickN      brownN          salmonN         LightSalmonN
orangeN         DarkOrangeN     coralN          tomatoN         OrangeRedN
redN            DeepPinkN       HotPinkN        pinkN           LightPinkN
PaleVioletRedN  maroonN         VioletRedN      magentaN        orchidN
plumN           MediumOrchidN   DarkOrchidN     purpleN         MediumPurpleN
thistleN
```

Escala de grises

Se puede obtener una escala de grises utilizando:

```
greyN
```

donde N es un número entre 0 y 100.

B.6 La tipografía Feta

Los símbolos siguientes están disponibles en la tipografía Emmentaler y es posible acceder a ellos directamente utilizando elementos de marcado textual como `g^\markup { \musicglyph #"scripts.segno" }`, véase Sección 1.8.2 [Formatear el texto], página 173.

```
space                           plus                            +
```

comma	,	hyphen	-
period	.	zero	0
one	1	two	2
three	3	four	4
five	5	six	6
seven	7	eight	8
nine	9	f	*f*
m	*m*	p	*p*
r	*r*	s	*s*
z	*z*	rests.0	𝄻
rests.1	𝄺	rests.0o	𝄼
rests.1o	𝄽	rests.M3	𝄩
rests.M2	𝄨	rests.M1	𝄧
rests.2	𝄾	rests.2classical	𝄽
rests.3	𝄿	rests.4	𝅀

rests.5		rests.6	
rests.7		accidentals.sharp	
accidentals .sharp.arrowup		accidentals .sharp.arrowdown	
accidentals .sharp.arrowboth		accidentals.sharp .slashslash.stem	
accidentals.sharp .slashslashslash.stemstem		accidentals.sharp .slashslashslash.stem	
accidentals.sharp .slashslash.stemstemstem		accidentals.natural	
accidentals .natural.arrowup		accidentals .natural.arrowdown	
accidentals .natural.arrowboth		accidentals.flat	
accidentals.flat.arrowup		accidentals .flat.arrowdown	
accidentals .flat.arrowboth		accidentals.flat.slash	
accidentals.flat .slashslash		accidentals .mirroredflat.flat	
accidentals.mirroredflat		accidentals .mirroredflat.backslash	
accidentals.flatflat		accidentals .flatflat.slash	
accidentals.doublesharp		accidentals.rightparen	
accidentals.leftparen		arrowheads.open.01	

arrowheads.open.0M1		arrowheads.open.11	
arrowheads.open.1M1		arrowheads.close.01	
arrowheads.close.0M1		arrowheads.close.11	
arrowheads.close.1M1		dots.dot	
noteheads.uM2		noteheads.dM2	
noteheads.sM1		noteheads.s0	
noteheads.s1		noteheads.s2	
noteheads.s0diamond		noteheads.s1diamond	
noteheads.s2diamond		noteheads.s0triangle	
noteheads.d1triangle		noteheads.u1triangle	
noteheads.u2triangle		noteheads.d2triangle	
noteheads.s0slash		noteheads.s1slash	
noteheads.s2slash		noteheads.s0cross	
noteheads.s1cross		noteheads.s2cross	
noteheads.s2xcircle		noteheads.s0do	

noteheads.d1do		noteheads.u1do	
noteheads.d2do		noteheads.u2do	
noteheads.s0re		noteheads.u1re	
noteheads.d1re		noteheads.u2re	
noteheads.d2re		noteheads.s0mi	
noteheads.s1mi		noteheads.s2mi	
noteheads.u0fa		noteheads.d0fa	
noteheads.u1fa		noteheads.d1fa	
noteheads.u2fa		noteheads.d2fa	
noteheads.s0la		noteheads.s1la	
noteheads.s2la		noteheads.s0ti	
noteheads.u1ti		noteheads.d1ti	
noteheads.u2ti		noteheads.d2ti	
scripts.ufermata		scripts.dfermata	
scripts.ushortfermata		scripts.dshortfermata	

scripts.ulongfermata		scripts.dlongfermata	
scripts.uverylongfermata		scripts.dverylongfermata	
scripts.thumb		scripts.sforzato	
scripts.espr		scripts.staccato	
scripts.ustaccatissimo		scripts.dstaccatissimo	
scripts.tenuto		scripts.uportato	
scripts.dportato		scripts.umarcato	
scripts.dmarcato		scripts.open	
scripts.stopped		scripts.upbow	
scripts.downbow		scripts.reverseturn	
scripts.turn		scripts.trill	
scripts.upedalheel		scripts.dpedalheel	
scripts.upedaltoe		scripts.dpedaltoe	
scripts.flageolet		scripts.segno	
scripts.coda		scripts.varcoda	

scripts.rcomma

scripts.lcomma

scripts.rvarcomma 

scripts.lvarcomma 

scripts.arpeggio 

scripts.trill_element

scripts.arpeggio.arrow.M1 

scripts.arpeggio.arrow.1 

scripts.trilelement

scripts.prall

scripts.mordent 

scripts.prallprall

scripts.prallmordent

scripts.upprall

scripts.upmordent

scripts.pralldown

scripts.downprall

scripts.downmordent

scripts.prallup

scripts.lineprall

scripts.caesura.curved

scripts.caesura.straight

flags.u3 

flags.u4 

flags.u5 

flags.u6 

flags.u7 

flags.d3 

flags.ugrace		flags.dgrace	
flags.d4		flags.d5	
flags.d6		flags.d7	
clefs.C		clefs.C_change	
clefs.F		clefs.F_change	
clefs.G		clefs.G_change	
clefs.percussion		clefs.percussion_change	
clefs.tab		clefs.tab_change	
timesig.C44		timesig.C22	
pedal.*		pedal.M	
pedal..	.	pedal.P	
pedal.d		pedal.e	
pedal.Ped		brackettips.up	
brackettips.down		accordion.accDiscant	

accordion.accDot

accordion.accFreebase 

accordion.accStdbase 

accordion.accBayanbase 

accordion.accOldEE 

rests.M3neomensural 

rests.M2neomensural

rests.M1neomensural

rests.0neomensural

rests.1neomensural

rests.2neomensural

rests.3neomensural

rests.4neomensural

rests.M3mensural

rests.M2mensural

rests.M1mensural

rests.0mensural

rests.1mensural

rests.2mensural

rests.3mensural

rests.4mensural

noteheads.s1neomensural 

noteheads.sM3neomensural 

noteheads.sM2neomensural 

noteheads.sM1neomensural 

noteheads.s0harmonic 

noteheads.s2harmonic 

noteheads.s0neomensural 

noteheads.s1neomensural

noteheads.s2neomensural 

noteheads.s1mensural 

noteheads.sM3mensural 

noteheads.sM2mensural 

noteheads.sM1mensural 

noteheads.s0mensural

noteheads.s1mensural

noteheads.s2mensural

noteheads.s0petrucci 

noteheads.s1petrucci 

noteheads.s2petrucci 

noteheads
.svaticana.punctum

noteheads.svaticana
.punctum.cavum

noteheads.svaticana
.linea.punctum

noteheads.svaticana
.linea.punctum.cavum

noteheads.svaticana
.inclinatum

noteheads.svaticana.lpes

noteheads
.svaticana.vlpes

noteheads.svaticana.upes

noteheads
.svaticana.vupes

noteheads
.svaticana.plica

noteheads
.svaticana.vplica

noteheads
.svaticana.epiphonus

noteheads.svaticana
.vepiphonus

noteheads.svaticana
.reverse.plica

noteheads.svaticana
.reverse.vplica

noteheads.svaticana
.inner.cephalicus

noteheads.svaticana
.cephalicus

noteheads
.svaticana.quilisma

noteheads.ssolesmes
.incl.parvum

noteheads
.ssolesmes.auct.asc

noteheads .ssolesmes.auct.desc		noteheads.ssolesmes .incl.auctum	
noteheads .ssolesmes.stropha		noteheads.ssolesmes .stropha.aucta	
noteheads .ssolesmes.oriscus		noteheads.smedicaea .inclinatum	
noteheads .smedicaea.punctum		noteheads .smedicaea.rvirga	
noteheads .smedicaea.virga		noteheads .shufnagel.punctum	
noteheads .shufnagel.virga		noteheads.shufnagel.lpes	
clefs.vaticana.do		clefs.vaticana.do_change	
clefs.vaticana.fa		clefs.vaticana.fa_change	
clefs.medicaea.do		clefs.medicaea.do_change	
clefs.medicaea.fa		clefs.medicaea.fa_change	
clefs.neomensural.c		clefs.neomensural .c_change	
clefs.petrucci.c1		clefs.petrucci.c1_change	
clefs.petrucci.c2		clefs.petrucci.c2_change	
clefs.petrucci.c3		clefs.petrucci.c3_change	
clefs.petrucci.c4		clefs.petrucci.c4_change	

clefs.petrucci.c5		clefs.petrucci.c5_change	
clefs.mensural.c		clefs.mensural.c_change	
clefs.petrucci.f		clefs.petrucci.f_change	
clefs.mensural.f		clefs.mensural.f_change	
clefs.petrucci.g		clefs.petrucci.g_change	
clefs.mensural.g		clefs.mensural.g_change	
clefs.hufnagel.do		clefs.hufnagel.do_change	
clefs.hufnagel.fa		clefs.hufnagel.fa_change	
clefs.hufnagel.do.fa		clefs.hufnagel .do.fa_change	
custodes.hufnagel.u0		custodes.hufnagel.u1	
custodes.hufnagel.u2		custodes.hufnagel.d0	
custodes.hufnagel.d1		custodes.hufnagel.d2	
custodes.medicaea.u0		custodes.medicaea.u1	
custodes.medicaea.u2		custodes.medicaea.d0	
custodes.medicaea.d1		custodes.medicaea.d2	

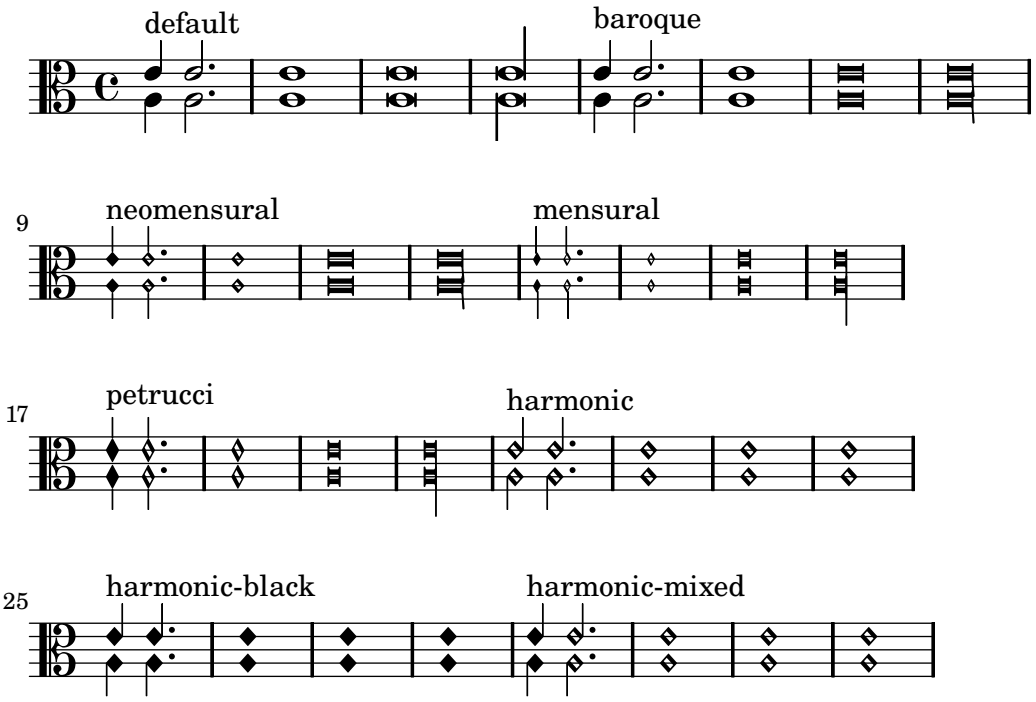
custodes.vaticana.u0		custodes.vaticana.u1	
custodes.vaticana.u2		custodes.vaticana.d0	
custodes.vaticana.d1		custodes.vaticana.d2	
custodes.mensural.u0		custodes.mensural.u1	
custodes.mensural.u2		custodes.mensural.d0	
custodes.mensural.d1		custodes.mensural.d2	
accidentals.medicaeaM1		accidentals.vaticanaM1	
accidentals.vaticana0		accidentals.mensural1	
accidentals.mensuralM1		accidentals.hufnagelM1	
flags.mensuralu03		flags.mensuralu13	
flags.mensuralu23		flags.mensurald03	
flags.mensurald13		flags.mensurald23	
flags.mensuralu04		flags.mensuralu14	
flags.mensuralu24		flags.mensurald04	
flags.mensurald14		flags.mensurald24	

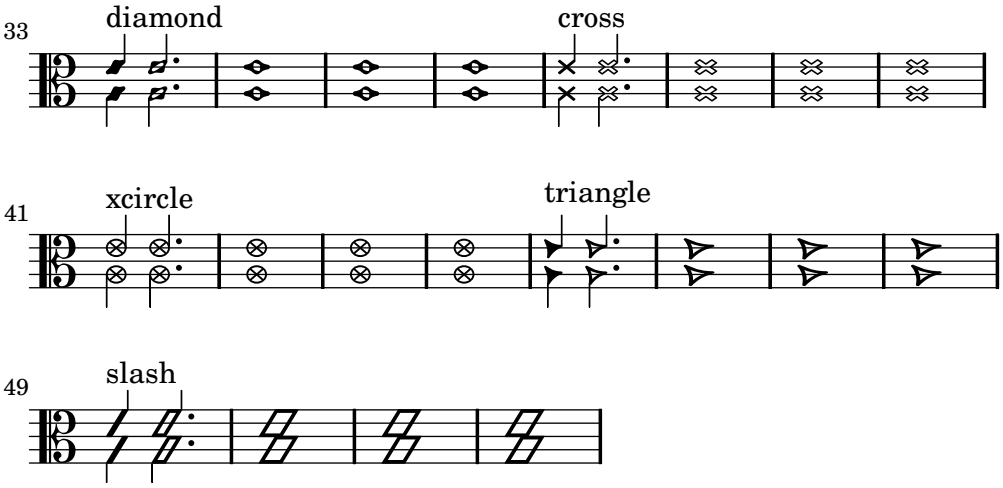
flags.mensuralu05		flags.mensuralu15	
flags.mensuralu25		flags.mensurald05	
flags.mensurald15		flags.mensurald25	
flags.mensuralu06		flags.mensuralu16	
flags.mensuralu26		flags.mensurald06	
flags.mensurald16		flags.mensurald26	
timesig.mensural44		timesig.mensural22	
timesig.mensural32		timesig.mensural64	
timesig.mensural94		timesig.mensural34	
timesig.mensural68		timesig.mensural98	
timesig.mensural48		timesig.mensural68alt	
timesig.mensural24		timesig.neomensural44	
timesig.neomensural22		timesig.neomensural32	
timesig.neomensural64		timesig.neomensural94	

timesig.neomensural34	𝅜	timesig.neomensural68	𝅝
timesig.neomensural98	𝅜	timesig.neomensural48	𝅝
timesig.neomensural68alt	𝅝	timesig.neomensural24	𝅜
scripts.ictus	ˌ	scripts.uaccentus	ˏ
scripts.daccentus	ˎ	scripts.usemicirculus	˓
scripts.dsemicirculus	ˑ	scripts.circulus	˚
scripts.augmentum	.	scripts .usignumcongruentiae	𝄐
scripts .dsignumcongruentiae	𝄐	dots.dotvaticana	.

B.7 Estilos de cabezas de nota

Se pueden usar los siguientes estilos para las cabezas de las notas.





B.8 Text markup commands

The following commands can all be used inside `\markup { }`.

B.8.1 Font

`\abs-fontsize` *size* (number) *arg* (markup)

Use *size* as the absolute font size to display *arg*. Adjusts `baseline-skip` and `word-space` accordingly.

```
\markup {
  default text font size
  \hspace #2
  \abs-fontsize #16 { text font size 16 }
  \hspace #2
  \abs-fontsize #12 { text font size 12 }
}
```

default text font size text font size 16 text font size 12

`\bold` *arg* (markup)

Switch to bold font-series.

```
\markup {
  default
  \hspace #2
  \bold
  bold
}
```

default **bold**

`\box` *arg* (markup)

Draw a box round *arg*. Looks at `thickness`, `box-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \override #'(box-padding . 0.5)
  \box
  \line { V. S. }
}
```

V. S.

Used properties:

- `box-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\caps` *arg* (markup)

Copy of the `\smallCaps` command.

```
\markup {
  default
  \hspace #2
  \caps {
    Text in small caps
  }
}
```

default TEXT IN SMALL CAPS

`\dynamic` *arg* (markup)

Use the dynamic font. This font only contains **s**, **f**, **m**, **z**, **p**, and **r**. When producing phrases, like 'più **f**', the normal words (like 'più') should be done in a different font. The recommended font for this is bold and italic.

```
\markup {
  \dynamic {
    sfzp
  }
}
```

sfzp

`\finger` *arg* (markup)

Set *arg* as small numbers.

```
\markup {
  \finger {
    1 2 3 4 5
  }
}
```

1 2 3 4 5

`\fontCaps` *arg* (markup)

Set `font-shape` to `caps`

Note: `\fontCaps` requires the installation and selection of fonts which support the `caps` font shape.

`\fontsize` *increment* (number) *arg* (markup)

Add *increment* to the font-size. Adjusts `baseline-skip` accordingly.

```
\markup {
  default
  \hspace #2
  \fontsize #-1.5
  smaller
```

```
}
```

default smaller

Used properties:

- `baseline-skip` (2)
- `word-space` (1)
- `font-size` (0)

`\huge` *arg* (markup)
Set font size to +2.

```
\markup {
  default
  \hspace #2
  \huge
  huge
}
```

default huge

`\italic` *arg* (markup)
Use italic `font-shape` for *arg*.

```
\markup {
  default
  \hspace #2
  \italic
  italic
}
```

default *italic*

`\large` *arg* (markup)
Set font size to +1.

```
\markup {
  default
  \hspace #2
  \large
  large
}
```

default large

`\larger` *arg* (markup)
Increase the font size relative to the current setting.

```
\markup {
  default
  \hspace #2
  \larger
  larger
}
```

default larger

`\magnify` *sz* (number) *arg* (markup)
Set the font magnification for its argument. In the following example, the middle A is 10% larger:

```
A \magnify #1.1 { A } A
```

Note: Magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.

```
\markup {
  default
  \hspace #2
  \magnify #1.5 {
    50% larger
  }
}
```

default 50% larger

`\medium` *arg* (markup)
Switch to medium font-series (in contrast to bold).

```
\markup {
  \bold {
    some bold text
    \hspace #2
    \medium {
      medium font series
    }
    \hspace #2
    bold again
  }
}
```

some bold text medium font series **bold again**

`\normal-size-sub` *arg* (markup)
Set *arg* in subscript with a normal font size.

```
\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}
```

default $_{\text{subscript in standard size}}$

Used properties:

- `baseline-skip`

`\normal-size-super` *arg* (markup)
Set *arg* in superscript with a normal font size.

```
\markup {
  default
  \normal-size-super {
    superscript in standard size
  }
}
```

default superscript in standard size

Used properties:

- `baseline-skip`

`\normal-text` *arg* (markup)
Set all font related properties (except the size) to get the default normal text font, no matter what font was used earlier.

```
\markup {
  \huge \bold \sans \caps {
    Some text with font overrides
    \hspace #2
    \normal-text {
      Default text, same font-size
    }
    \hspace #2
    More text as before
  }
}
```

SOME TEXT WITH FONT OVERRIDES Default text, same font-size **MORE TEX**

`\normalsize` *arg* (markup)
Set font size to default.

```
\markup {
  \teeny {
    this is very small
    \hspace #2
    \normalsize {
      normal size
    }
    \hspace #2
    teeny again
  }
}
```

this is very small normal size teeny again

`\number` *arg* (markup)
Set font family to `number`, which yields the font used for time signatures and fingerings. This font contains numbers and some punctuation; it has no letters.

```
\markup {
  \number {
    0 1 2 3 4 5 6 7 8 9 . ,
```

```
  }
}
```

0123456789.,

`\roman` *arg* (markup)
Set font family to `roman`.

```
\markup {
  \sans \bold {
    sans serif, bold
    \hspace #2
    \roman {
      text in roman font family
    }
    \hspace #2
    return to sans
  }
}
```

sans serif, bold text in roman font family return to sans

`\sans` *arg* (markup)
Switch to the sans serif font family.

```
\markup {
  default
  \hspace #2
  \sans {
    sans serif
  }
}
```

default sans serif

`\simple` *str* (string)
A simple text string; `\markup { foo }` is equivalent with `\markup { \simple #"foo" }`.

Note: for creating standard text markup or defining new markup commands, the use of `\simple` is unnecessary.

```
\markup {
  \simple #"simple"
  \simple #"text"
  \simple #"strings"
}
```

simple text strings

`\small` *arg* (markup)
Set font size to -1.

```
\markup {
  default
  \hspace #2
```

```
  \small
  small
}
```

default small

`\smallCaps` *arg* (markup)
Emit *arg* as small caps.

Note: `\smallCaps` does not support accented characters.

```
\markup {
  default
  \hspace #2
  \smallCaps {
    Text in small caps
  }
}
```

default TEXT IN SMALL CAPS

`\smaller` *arg* (markup)
Decrease the font size relative to the current setting.

```
\markup {
  \fontsize #3.5 {
    some large text
    \hspace #2
    \smaller {
      a bit smaller
    }
    \hspace #2
    more large text
  }
}
```

some large text a bit smaller more large text

`\sub` *arg* (markup)
Set *arg* in subscript.

```
\markup {
  \concat {
    H
    \sub {
      2
    }
    O
  }
}
```

H_2O

Used properties:

- `baseline-skip`
- `font-size` (0)

`\super` *arg* (markup)
Set *arg* in superscript.

```
\markup {
  E =
  \concat {
    mc
    \super
    2
  }
}
```

$E = mc^{2}$

Used properties:

- `baseline-skip`
- `font-size` (0)

`\teeny` *arg* (markup)
Set font size to -3.

```
\markup {
  default
  \hspace #2
  \teeny
  teeny
}
```

default teeny

`\text` *arg* (markup)
Use a text font instead of music symbol or music alphabet font.

```
\markup {
  \number {
    1, 2,
    \text {
      three, four,
    }
    5
  }
}
```

1, 2, three, four, **5**

`\tiny` *arg* (markup)
Set font size to -2.

```
\markup {
  default
  \hspace #2
  \tiny
```

```
  tiny
}
```

default tiny

`\typewriter` *arg* (markup)
Use `font-family` typewriter for *arg*.

```
\markup {
  default
  \hspace #2
  \typewriter
  typewriter
}
```

default `typewriter`

`\underline` *arg* (markup)
Underline *arg*. Looks at `thickness` to determine line thickness and y-offset.

```
\markup {
  default
  \hspace #2
  \override #'(thickness . 2)
  \underline {
    underline
  }
}
```

default underline

Used properties:

- `thickness` (1)

`\upright` *arg* (markup)
Set `font-shape` to `upright`. This is the opposite of `italic`.

```
\markup {
  \italic {
    italic text
    \hspace #2
    \upright {
      upright text
    }
    \hspace #2
    italic again
  }
}
```

italic text upright text *italic again*

B.8.2 Align

`\center-align` *arg* (markup)
Align `arg` to its X center.

```
\markup {
  \column {
    one
    \center-align
    two
    three
  }
}
```

one
two
three

`\center-column` *args* (list of markups)
Put `args` in a centered column.

```
\markup {
  \center-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\column` *args* (list of markups)
Stack the markups in *args* vertically. The property `baseline-skip` determines the space between markups in *args*.

```
\markup {
  \column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\combine` *arg1* (markup) *arg2* (markup)
Print two markups on top of each other.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; the follow example will not compile:

```
\combine { a list }
```

```
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \combine
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
}
```



`\concat` *args* (list of markups)
Concatenate *args* in a horizontal line, without spaces in between. Strings and simple markups are concatenated on the input level, allowing ligatures. For example, `\concat { "f" \simple #"i" }` is equivalent to `"fi"`.

```
\markup {
  \concat {
    one
    two
    three
  }
}
```

onetwothree

`\dir-column` *args* (list of markups)
Make a column of *args*, going up or down, depending on the setting of the `direction` layout property.

```
\markup {
  \override #`(direction . ,UP) {
    \dir-column {
      going up
    }
  }
  \hspace #1
  \dir-column {
    going down
  }
  \hspace #1
  \override #'(direction . 1) {
    \dir-column {
      going up
    }
  }
}
```

up up
going going going
down

Used properties:

- `baseline-skip`
- `direction`

`\fill-line` *args* (list of markups)
Put *markups* in a horizontal line of width *line-width*. The markups are spaced or flushed to fill the entire line. If there are no arguments, return an empty stencil.

```
\markup {
  \column {
    \fill-line {
      Words evenly spaced across the page
    }
    \null
    \fill-line {
      \line { Text markups }
      \line {
        \italic { evenly spaced }
      }
      \line { across the page }
    }
  }
}
```

Words evenly spaced across the page

Text markups *evenly spaced* across the page

Used properties:

- `line-width` (#f)
- `word-space` (1)
- `text-direction` (1)

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)
Align *arg* in *axis* direction to the *dir* side.

```
\markup {
  \column {
    one
    \general-align #X #LEFT
    two
    three
    \null
    one
    \general-align #X #CENTER
    two
    three
    \null
    \line {
      one
      \general-align #Y #UP
      two
```

```
      three
    }
    \null
    \line {
      one
      \general-align #Y #3.2
      two
      three
    }
  }
}
```

one
two
three

one
two
three

one two three

one three
two

`\halign` *dir* (number) *arg* (markup)
: Set horizontal alignment. If *dir* is `-1`, then it is left-aligned, while `+1` is right. Values in between interpolate alignment accordingly.

```
\markup {
  \column {
    one
    \halign #LEFT
    two
    three
    \null
    one
    \halign #CENTER
    two
    three
    \null
    one
    \halign #RIGHT
    two
    three
    \null
    one
    \halign #-5
    two
    three
  }
}
```

one
two
three

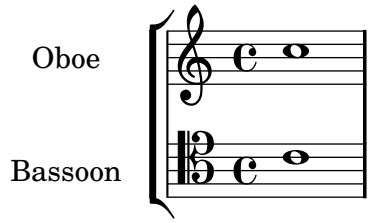
one
two
three

one
two
three

one
two
three

`\hcenter-in` *length* (number) *arg* (markup)
Center *arg* horizontally within a box of extending *length*/2 to the left and right.

```
\new StaffGroup <<
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Oboe
    }
    c''1
  }
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Bassoon
    }
    \clef tenor
    c'1
  }
>>
```



`\hspace` *amount* (number)
Create an invisible object taking up horizontal space *amount*.

```
\markup {
  one
  \hspace #2
  two
  \hspace #8
  three
}
```

one two three

`\justify-field` *symbol* (symbol)
: Justify the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  description = "Lorem ipsum dolor sit amet, consectetur adipisicing
    elit, sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua.  Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \justify-field #'header:description
    }
  }
}

\markup {
  \null
}
```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\justify` *args* (list of markups)
: Like `\wordwrap`, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; X is the number of staff spaces.

```
\markup {
  \justify {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\justify-string` *arg* (string)

Justify a string. Paragraphs may be separated with double newlines

```
\markup {
  \override #'(line-width . 40)
  \justify-string #"Lorem ipsum dolor sit amet, consectetur
      adipisicing elit, sed do eiusmod tempor incididunt ut labore
      et dolore magna aliqua.


      Ut enim ad minim veniam, quis nostrud exercitation ullamco
      laboris nisi ut aliquip ex ea commodo consequat.


      Excepteur sint occaecat cupidatat non proident, sunt in culpa
      qui officia deserunt mollit anim id est laborum"
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

`\left-align` *arg* (markup)

Align *arg* on its left edge.

```
\markup {
  \column {
    one
    \left-align
    two
    three
  }
}
```

one
two
three

`\left-column` *args* (list of markups)
Put `args` in a left-aligned column.

```
\markup {
  \left-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\line` *args* (list of markups)
Put *args* in a horizontal line. The property `word-space` determines the space between markups in *args*.

```
\markup {
  \line {
    one two three
  }
}
```

one two three

Used properties:

- `text-direction` (1)
- `word-space`

`\lower` *amount* (number) *arg* (markup)
Lower *arg* by the distance *amount*. A negative *amount* indicates raising; see also `\raise`.

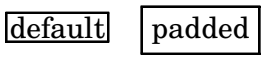
```
\markup {
  one
  \lower #3
  two
  three
}
```

one three
two

`\pad-around` *amount* (number) *arg* (markup)
Add padding *amount* all around *arg*.

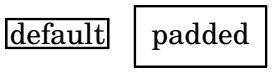
```
\markup {
  \box {
    default
```

```
  }
  \hspace #2
  \box {
    \pad-around #0.5 {
      padded
    }
  }
}
```



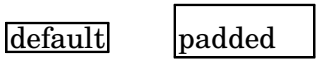
`\pad-markup` *amount* (number) *arg* (markup)
Add space around a markup object.

```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-markup #1 {
      padded
    }
  }
}
```



`\pad-to-box` *x-ext* (pair of numbers) *y-ext* (pair of numbers) *arg* (markup)
Make *arg* take at least *x-ext*, *y-ext* space.

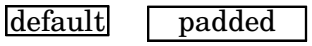
```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-to-box #'(0 . 10) #'(0 . 3) {
      padded
    }
  }
}
```



`\pad-x` *amount* (number) *arg* (markup)
Add padding *amount* around *arg* in the X direction.

```
\markup {
  \box {
    default
  }
```

```
  \hspace #4
  \box {
    \pad-x #2 {
      padded
    }
  }
}
```



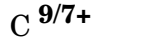
`\put-adjacent` *axis* (integer) *dir* (direction) *arg1* (markup) *arg2* (markup)
Put *arg2* next to *arg1*, without moving *arg1*.

`\raise` *amount* (number) *arg* (markup)
Raise *arg* by the distance *amount*. A negative *amount* indicates lowering, see also `\lower`.

The argument to `\raise` is the vertical displacement amount, measured in (global) staff spaces. `\raise` and `\super` raise objects in relation to their surrounding markups.

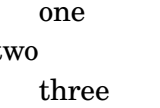
If the text object itself is positioned above or below the staff, then `\raise` cannot be used to move it, since the mechanism that positions it next to the staff cancels any shift made with `\raise`. For vertical positioning, use the `padding` and/or `extra-offset` properties.

```
\markup {
  C
  \small
  \bold
  \raise #1.0
  9/7+
}
```



`\right-align` *arg* (markup)
Align *arg* on its right edge.

```
\markup {
  \column {
    one
    \right-align
    two
    three
  }
}
```



`\right-column` *args* (list of markups)
Put `args` in a right-aligned column.

```
\markup {
  \right-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\rotate` *ang* (number) *arg* (markup)
: Rotate object with *ang* degrees around its center.

```
\markup {
  default
  \hspace #2
  \rotate #45
  \line {
    rotated 45˚
  }
}
```



`\translate` *offset* (pair of numbers) *arg* (markup)
: Translate *arg* relative to its surroundings. *offset* is a pair of numbers representing the displacement in the X and Y axis.

```
\markup {
  *
  \translate #'(2 . 3)
  \line { translated two spaces right, three up }
}
```

translated two spaces right, three up
*

`\translate-scaled` *offset* (pair of numbers) *arg* (markup)
: Translate *arg* by *offset*, scaling the offset by the `font-size`.

```
\markup {
  \fontsize #5 {
    * \translate #'(2 . 3) translate
    \hspace #2
    * \translate-scaled #'(2 . 3) translate-scaled
  }
}
```

* translate * translate-scaled

Used properties:

- `font-size` (0)

`\vcenter` *arg* (markup)
Align `arg` to its Y center.

```
\markup {
  one
  \vcenter
  two
  three
}
```

one two three

`\wordwrap-field` *symbol* (symbol)
Wordwrap the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  description = "Lorem ipsum dolor sit amet, consectetur adipisicing
    elit, sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua.  Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \wordwrap-field #'header:descr
    }
  }
}

\markup {
  \null
}
```

My title

`\wordwrap` *args* (list of markups)
Simple wordwrap. Use `\override #'(line-width . X)` to set the line width, where X is the number of staff spaces.

```
\markup {
  \wordwrap {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
```

```
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-string` *arg* (string)

Wordwrap a string. Paragraphs may be separated with double newlines.

```
\markup {
  \override #'(line-width . 40)
  \wordwrap-string #"Lorem ipsum dolor sit amet, consectetur
      adipisicing elit, sed do eiusmod tempor incididunt ut labore
      et dolore magna aliqua.


      Ut enim ad minim veniam, quis nostrud exercitation ullamco
      laboris nisi ut aliquip ex ea commodo consequat.


      Excepteur sint occaecat cupidatat non proident, sunt in culpa
      qui officia deserunt mollit anim id est laborum"
}
```

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore
et dolore magna aliqua.
Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo
consequat.
Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia
deserunt mollit anim id est laborum

Used properties:

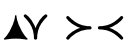
- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

B.8.3 Graphic

`\arrow-head` *axis* (integer) *dir* (direction) *filled* (boolean)

Produce an arrow head in specified direction and axis. Use the filled head if *filled* is specified.

```
\markup {
  \fontsize #5 {
    \general-align #Y #DOWN {
      \arrow-head #Y #UP ##t
      \arrow-head #Y #DOWN ##f
      \hspace #2
      \arrow-head #X #RIGHT ##f
      \arrow-head #X #LEFT ##f
    }
  }
}
```



`\beam` *width* (number) *slope* (number) *thickness* (number)

Create a beam with the specified parameters.

```
\markup {
  \beam #5 #1 #2
}
```



`\bracket` *arg* (markup)

Draw vertical brackets around *arg*.

```
\markup {
  \bracket {
    \note #"2." #UP
  }
}
```



`\circle` *arg* (markup)

Draw a circle around *arg*. Use `thickness`, `circle-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \circle {
    Hi
  }
}
```



Used properties:

- `circle-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\draw-circle` *radius* (number) *thickness* (number) *filled* (boolean)
A circle of radius *radius* and thickness *thickness*, optionally filled.

```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
}
```



`\draw-line` *dest* (pair of numbers)
A simple line.

```
\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}
```

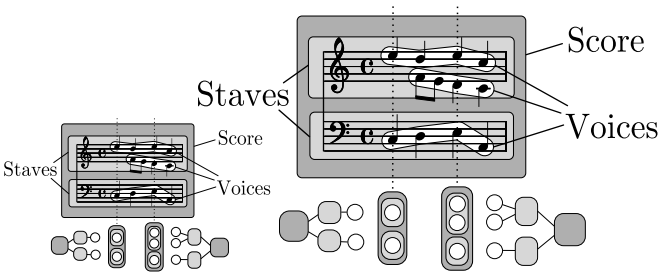


Used properties:

- `thickness` (1)

`\epsfile` *axis* (number) *size* (number) *file-name* (string)
Inline an EPS image. The image is scaled along *axis* to *size*.

```
\markup {
  \general-align #Y #DOWN {
    \epsfile #X #20 #"context-example.eps"
    \epsfile #Y #20 #"context-example.eps"
  }
}
```



`\filled-box` *xext* (pair of numbers) *yext* (pair of numbers) *blot* (number)
Draw a box with rounded corners of dimensions *xext* and *yext*. For example,

`\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0`

creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e., sharp corners).

```
\markup {
  \filled-box #'(0 . 4) #'(0 . 4) #0
  \filled-box #'(0 . 2) #'(-4 . 2) #0.4
  \filled-box #'(1 . 8) #'(0 . 7) #0.2
  \with-color #white
  \filled-box #'(-4.5 . -2.5) #'(3.5 . 5.5) #0.7
}
```



`\hbracket` *arg* (markup)
Draw horizontal brackets around *arg*.

```
\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}
```
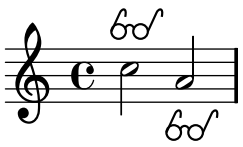
one two three

`\postscript` *str* (string)
This inserts *str* directly into the output as a PostScript command string.

```
eyeglassesps = #"
  0.15 setlinewidth
  -0.9 0 translate
  1.1 1.1 scale
  1.2 0.7 moveto
  0.7 0.7 0.5 0 361 arc
  stroke
  2.20 0.70 0.50 0 361 arc
  stroke
  1.45 0.85 0.30 0 180 arc
  stroke
  0.20 0.70 moveto
  0.80 2.00 lineto
  0.92 2.26 1.30 2.40 1.15 1.70 curveto
  stroke
  2.70 0.70 moveto
  3.30 2.00 lineto
  3.42 2.26 3.80 2.40 3.65 1.70 curveto
  stroke"

eyeglasses = \markup {
  \with-dimensions #'(0 . 4.4) #'(0 . 2.5)
  \postscript #eyeglassesps
}
```
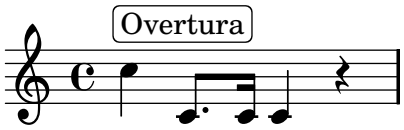
```
\relative c'' {
  c2^\eyeglasses
  a2_\eyeglasses
}
```



`\rounded-box` *arg* (markup)

Draw a box with rounded corners around *arg*. Looks at `thickness`, `box-padding` and `font-size` properties to determine line thickness and padding around the markup; the `corner-radius` property makes it possible to define another shape for the corners (default is 1).

```
c4^\markup {
  \rounded-box {
    Overtura
  }
}
c,8. c16 c4 r
```



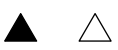
Used properties:

- `box-padding` (0.5)
- `font-size` (0)
- `corner-radius` (1)
- `thickness` (1)

`\triangle` *filled* (boolean)

A triangle, either filled or empty.

```
\markup {
  \triangle ##t
  \hspace #2
  \triangle ##f
}
```



Used properties:

- `baseline-skip` (2)
- `font-size` (0)
- `thickness` (0.1)

`\with-url` *url* (string) *arg* (markup)

Add a link to URL *url* around *arg*. This only works in the PDF backend.

```
\markup {
  \with-url #"http://lilypond.org/web/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}
```

LilyPond ... *music notation for everyone*

B.8.4 Music

`\doubleflat`

Draw a double flat symbol.

```
\markup {
  \doubleflat
}
```



`\doublesharp`

Draw a double sharp symbol.

```
\markup {
  \doublesharp
}
```


`\flat`

Draw a flat symbol.

```
\markup {
  \flat
}
```



`\musicglyph` *glyph-name* (string)

glyph-name is converted to a musical symbol; for example, `\musicglyph #"accidentals.natural"` selects the natural sign from the music font. See Sección "La tipografía Feta" in *Referencia de la Notación* for a complete listing of the possible glyphs.

```
\markup {
  \musicglyph #"f"
  \musicglyph #"rests.2"
  \musicglyph #"clefs.G_change"
}
```



`\natural`

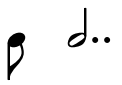
Draw a natural symbol.

```
\markup {
  \natural
}
```



`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)
: Construct a note symbol, with stem. By using fractional values for *dir*, longer or shorter stems can be obtained.

```
\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
}
```

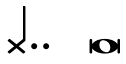


Used properties:

- `style` ('())
- `font-size` (0)

`\note` *duration* (string) *dir* (number)
: This produces a note with a stem pointing in *dir* direction, with the *duration* for the note head type and augmentation dots. For example, `\note #"4." #-0.75` creates a dotted quarter note, with a shortened down stem.

```
\markup {
  \override #'(style . cross) {
    \note #"4.." #UP
  }
  \hspace #2
  \note #"breve" #0
}
```



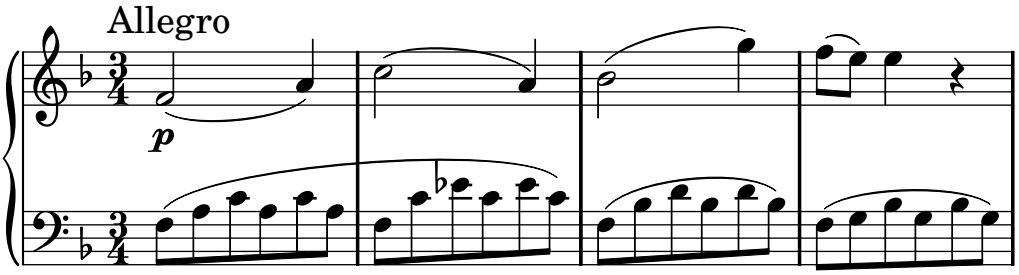
Used properties:

- `style` ('())
- `font-size` (0)

`\score` *score* (unknown)
: Inline an image of music.

```
\markup {
  \score {
    \new PianoStaff <<
      \new Staff \relative c' {
        \key f \major
        \time 3/4
        \mark \markup { Allegro }
        f2\p( a4)
        c2( a4)
```

```
            bes2( g'4)
            f8( e) e4 r
          }
          \new Staff \relative c {
            \clef bass
            \key f \major
            \time 3/4
            f8( a c a c a
            f c' es c es c)
            f,( bes d bes d bes)
            f( g bes g bes g)
          }
        >>
        \layout {
          indent = 0.0\cm
          \context {
            \Score
            \override RehearsalMark #'break-align-symbols =
              #'(time-signature key-signature)
            \override RehearsalMark #'self-alignment-X = #LEFT
          }
          \context {
            \Staff
            \override TimeSignature #'break-align-anchor-alignment = #LEFT
          }
        }
      }
    }
```



`\semiflat`

Draw a semiflat symbol.

```
\markup {
  \semiflat
}
```



`\semisharp`

Draw a semisharp symbol.

```
\markup {
  \semisharp
}
```



`\sesquiflat`
Draw a 3/2 flat symbol.

```
\markup {
  \sesquiflat
}
```



`\sesquisharp`
Draw a 3/2 sharp symbol.

```
\markup {
  \sesquisharp
}
```



`\sharp`
Draw a sharp symbol.

```
\markup {
  \sharp
}
```



`\tied-lyric` *str* (string)
Like simple-markup, but use tie characters for '~' tilde symbols.

```
\markup {
  \tied-lyric #"Lasciate~i monti"
}
```

Lasciate i monti

B.8.5 Instrument Specific Markup

`\fret-diagram` *definition-string* (string)
Make a (guitar) fret diagram. For example, say

```
\markup \fret-diagram #"s:0.75;6-x;5-x;4-o;3-2;2-3;1-2;"
```

for fret spacing 3/4 of staff space, D chord diagram

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
  - `s:`*number* – Set the fret spacing of the diagram (in staff spaces). Default: 1.
  - `t:`*number* – Set the line thickness (in staff spaces). Default: 0.05.
  - `h:`*number* – Set the height of the diagram in frets. Default: 4.
  - `w:`*number* – Set the width of the diagram in strings. Default: 6.
  - `f:`*number* – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.

- `d:`*number* – Set radius of dot, in terms of fret spacing. Default: 0.25.
- `p:`*number* – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
- `c:`*string1*`-`*string2*`-`*fret* – Include a barre mark from *string1* to *string2* on *fret*.
- *string*`-`*fret* – Place a dot on *string* at *fret*. If *fret* is '`o`', *string* is identified as open. If *fret* is '`x`', *string* is identified as muted.
- *string*`-`*fret*`-`*fingering* – Place a dot on *string* at *fret*, and label with *fingering* as defined by the `f:` code.

– Note: There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

Here is an example

```
\markup \fret-diagram-terse #"x;x;o;2;3;2;"
```

for a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.
- Mute strings are indicated by '`x`'.
- Open strings are indicated by '`o`'.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.
- Fingerings are given by following the fret number with a `-`, followed by the finger indicator, e.g. '`3-2`' for playing the third fret with the second finger.
- Where a barre indicator is desired, follow the fret (or fingering) symbol with `-(` to start a barre and `-)` to end the barre.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-verbose` *marking-list* (pair)

Make a fret diagram containing the symbols indicated in *marking-list*.

For example,

```
\markup \fret-diagram-verbose
  #'((mute 6) (mute 5) (open 4)
     (place-fret 3 2) (place-fret 2 3) (place-fret 1 2))
```

produces a standard D chord diagram without fingering indications.

Possible elements in *marking-list*:

`(mute` *`string-number`*`)`
: Place a small ‘x’ at the top of string *string-number*.

`(open` *`string-number`*`)`
: Place a small ‘o’ at the top of string *string-number*.

`(barre` *`start-string end-string fret-number`*`)`
: Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

`(capo` *`fret-number`*`)`
: Place a capo indicator (a large solid bar) across the entire fretboard at fret location *fret-number*. Also, set fret *fret-number* to be the lowest fret on the fret diagram.

`(place-fret` *`string-number fret-number finger-value`*`)`
: Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*. By default, the fret playing indicator is a solid dot. This can be changed by setting the value of the variable *dot-color*. If the *finger* part of the `place-fret` element is present, *finger-value* will be displayed according to the setting of the variable *finger-code*. There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\harp-pedal` *definition-string* (string)
: Make a harp pedal diagram.

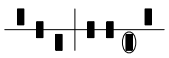
Possible elements in *definition-string*:

`^`	pedal is up
`-`	pedal is neutral
`v`	pedal is down
`|`	vertical divider line
`o`	the following pedal should be circled (indicating a change)

The function also checks if the string has the typical form of three pedals, then the divider and then the remaining four pedals. If not it prints out a warning. However, in any case, it will also print each symbol in the order as given. This means you can place the divider (even multiple dividers) anywhere you want, but you’ll have to live with the warnings.

The appearance of the diagram can be tweaked inter alia using the size property of the TextScript grob (`\override Voice.TextScript #'size = #0.3`) for the overall, the thickness property (`\override Voice.TextScript #'thickness = #3`) for the line thickness of the horizontal line and the divider. The remaining configuration (box sizes, offsets and spaces) is done by the harp-pedal-details list of properties (`\override Voice.TextScript #'harp-pedal-details #'box-width = #1`). It contains the following settings: `box-offset` (vertical shift of the box center for up/down pedals), `box-width`, `box-height`, `space-before-divider` (the spacing between two boxes before the divider) and `space-after-divider` (box spacing after the divider).

```
\markup \harp-pedal #"^-v|--ov^"
```



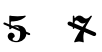
Used properties:

- `thickness` (0.5)
- `harp-pedal-details`
- `size` (1.0)

B.8.6 Other

`\backslashed-digit` *num* (integer)
A feta number, with backslash. This is for use in the context of figured bass notation.

```
\markup {
  \backslashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \backslashed-digit #7
}
```



Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\char` *num* (integer)
Produce a single character. Characters encoded in hexadecimal format require the prefix `#x`.

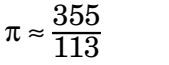
```
\markup {
  \char #65 \char ##x00a9
}
```



`\fraction` *arg1* (markup) *arg2* (markup)
Make a fraction of two markups.

```
\markup {

  \fraction 355 113
}
```



Used properties:

- `font-size` (0)

`\fromproperty` *symbol* (symbol)
Read the *symbol* from property settings, and produce a stencil from the markup contained within. If *symbol* is not defined, it returns an empty markup.

```
\header {
  myTitle = "myTitle"
  title = \markup {
    from
    \italic
    \fromproperty #'header:myTitle
  }
}
\markup {
  \null
}
```

from *myTitle*

`\lookup` *glyph-name* (string)
: Lookup a glyph by name.

```
\markup {
  \override #'(font-encoding . fetaBraces) {
    \lookup #"brace200"
    \hspace #2
    \rotate #180
    \lookup #"brace180"
  }
}
```



`\markalphabet` *num* (integer)
: Make a markup letter for *num*. The letters start with A to Z and continue with double letters.

```
\markup {
  \markalphabet #8
  \hspace #2
  \markalphabet #26
}
```

I AA

`\markletter` *num* (integer)
: Make a markup letter for *num*. The letters start with A to Z (skipping letter I), and continue with double letters.

```
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
```

J AB

`\null`

An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (symbol) *arg* (markup)

Apply the *procedure* markup command to *arg*. *procedure* should take a single argument.

`\override` *new-prop* (pair) *arg* (markup)

Add the argument *new-prop* to the property list. Properties may be any property supported by Sección "font-interface" in *Referencia de Funcionamiento Interno*, Sección "text-interface" in *Referencia de Funcionamiento Interno* and Sección "instrument-specific-markup-interface" in *Referencia de Funcionamiento Interno*.

```
\markup {
  \line {
    \column {
      default
      baseline-skip
    }
    \hspace #2
    \override #'(baseline-skip . 4) {
      \column {
        increased
        baseline-skip
      }
    }
  }
}
```

default baseline-skip increased baseline-skip

`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)

Reference to a page number. *label* is the label set on the referenced page (using the `\label` command), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

`\slashed-digit` *num* (integer)

A feta number, with slash. This is for use in the context of figured bass notation.

```
\markup {
  \slashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \slashed-digit #7
}
```

5 7

Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\stencil` *stil* (unknown)
Use a stencil as markup.

```
\markup {
  \stencil #(make-circle-stencil 2 0 #t)
}
```



`\strut`
Create a box of the same height as the space in the current font.

`\transparent` *arg* (markup)
Make *arg* transparent.

```
\markup {
  \transparent {
    invisible text
  }
}
```

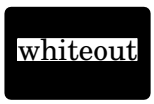
`\verbatim-file` *name* (string)
Read the contents of file *name*, and include it verbatim.

```
\markup {
  \verbatim-file #"simple.ly"
}
```

```
%% A simple piece in LilyPond, a scale.
\relative c' {
  c d e f g a b c
}
%% Optional helper for automatic updating by convert-ly.  May be omit
\version "2.12.0"
```

`\whiteout` *arg* (markup)
Provide a white background for *arg*.

```
\markup {
  \combine
    \filled-box #'(-1 . 10) #'(-3 . 4) #1
    \whiteout whiteout
}
```



`\with-color` *color* (list) *arg* (markup)
Draw *arg* in color specified by *color*.

```
\markup {
  \with-color #red
  red
  \hspace #2
  \with-color #green
  green
  \hspace #2
  \with-color #blue
  blue
}
```

red green blue

`\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)
Set the dimensions of *arg* to *x* and *y*.

B.9 Text markup list commands

The following commands can all be used with `\markuplines`.

`\column-lines` *args* (list of markups)
Like `\column`, but return a list of lines instead of a single markup. `baseline-skip` determines the space between each markup in *args*.

Used properties:

- `baseline-skip`

`\justified-lines` *args* (list of markups)
Like `\justify`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\override-lines` *new-prop* (pair) *args* (list of markups)
Like `\override`, for markup lists.

`\wordwrap-internal` *justify* (boolean) *args* (list of markups)
Internal markup list command used to define `\justify` and `\wordwrap`.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)

`\wordwrap-lines` *args* (list of markups)
Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`

- `line-width` (#f)
- `baseline-skip`

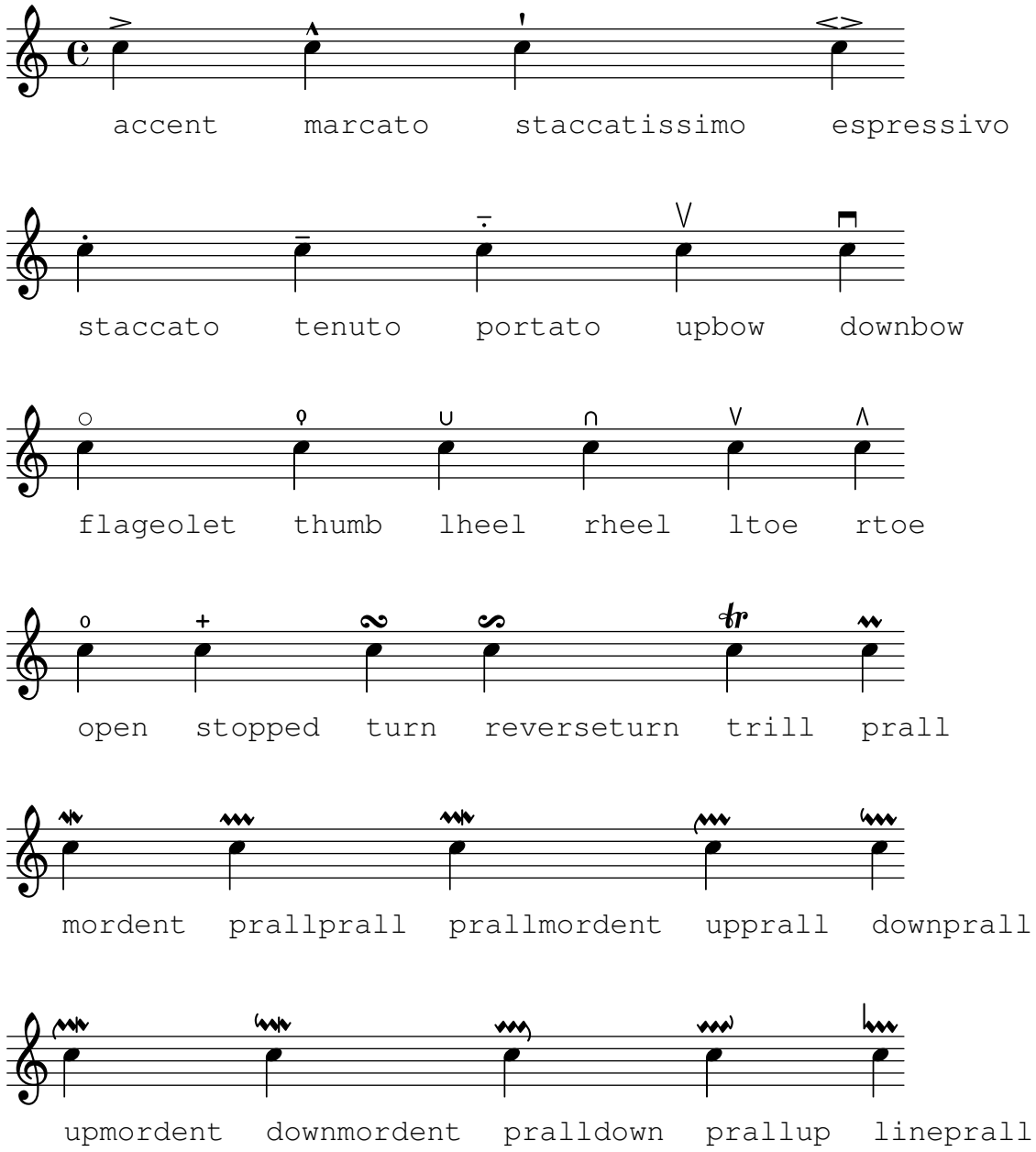
`\wordwrap-string-internal` *justify* (boolean) *arg* (string)
Internal markup list command used to define `\justify-string` and `\wordwrap-string`.

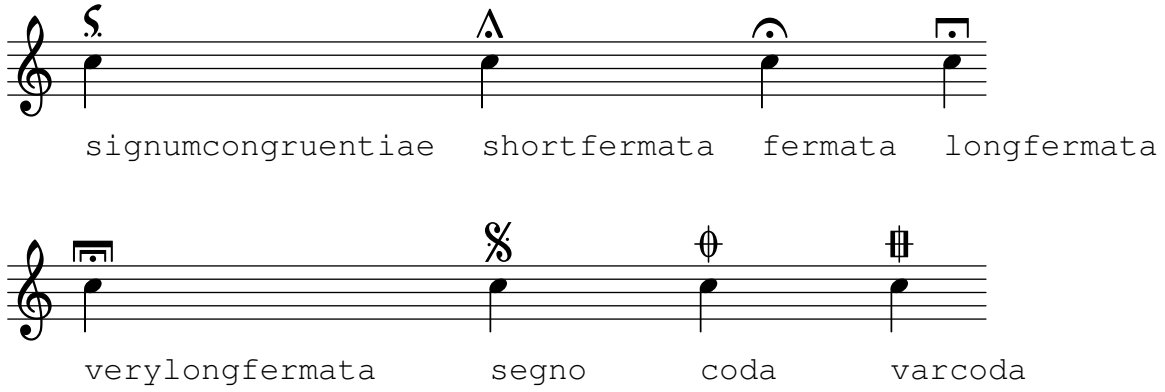
Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`

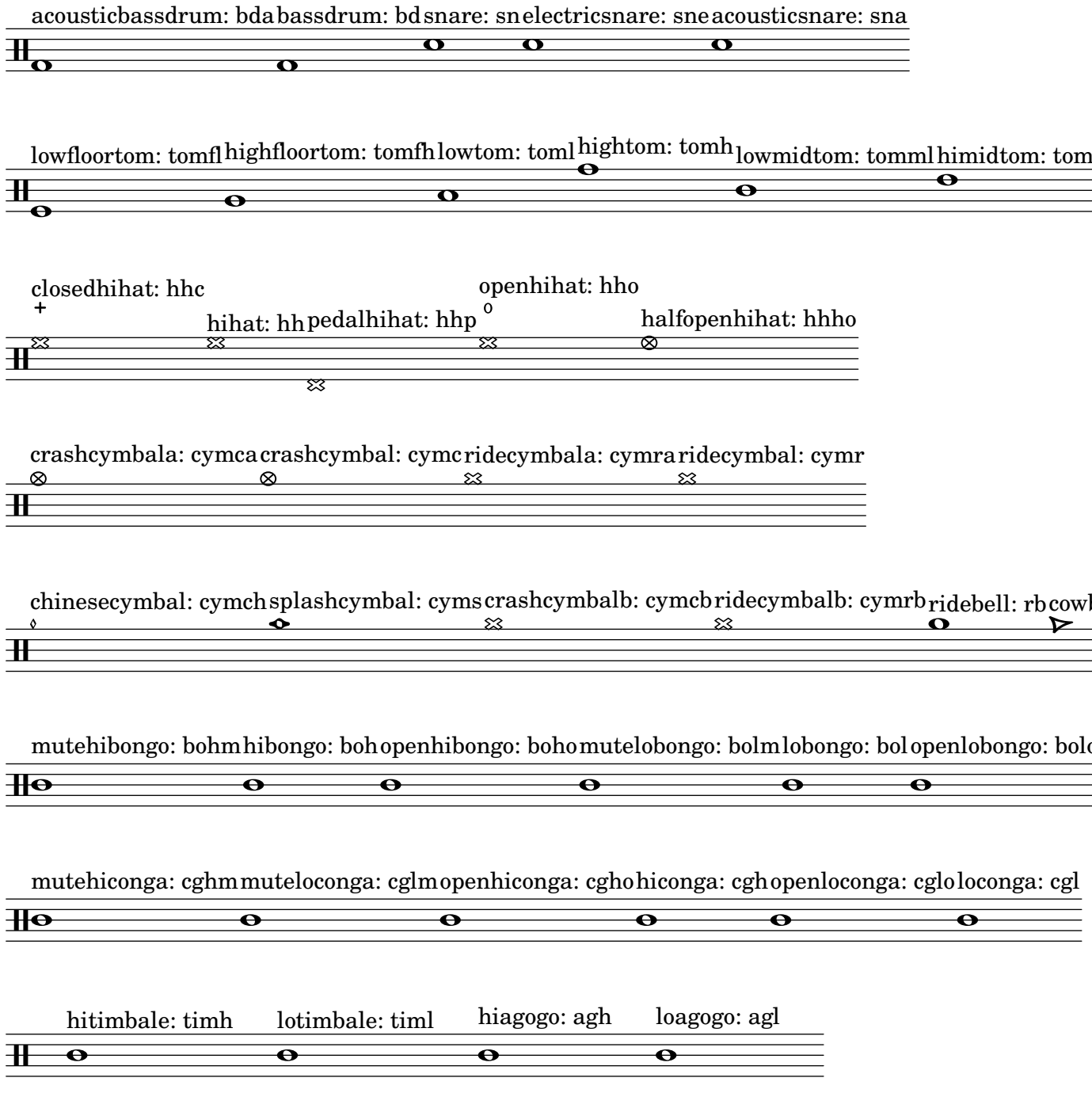
B.10 Lista de articulaciones

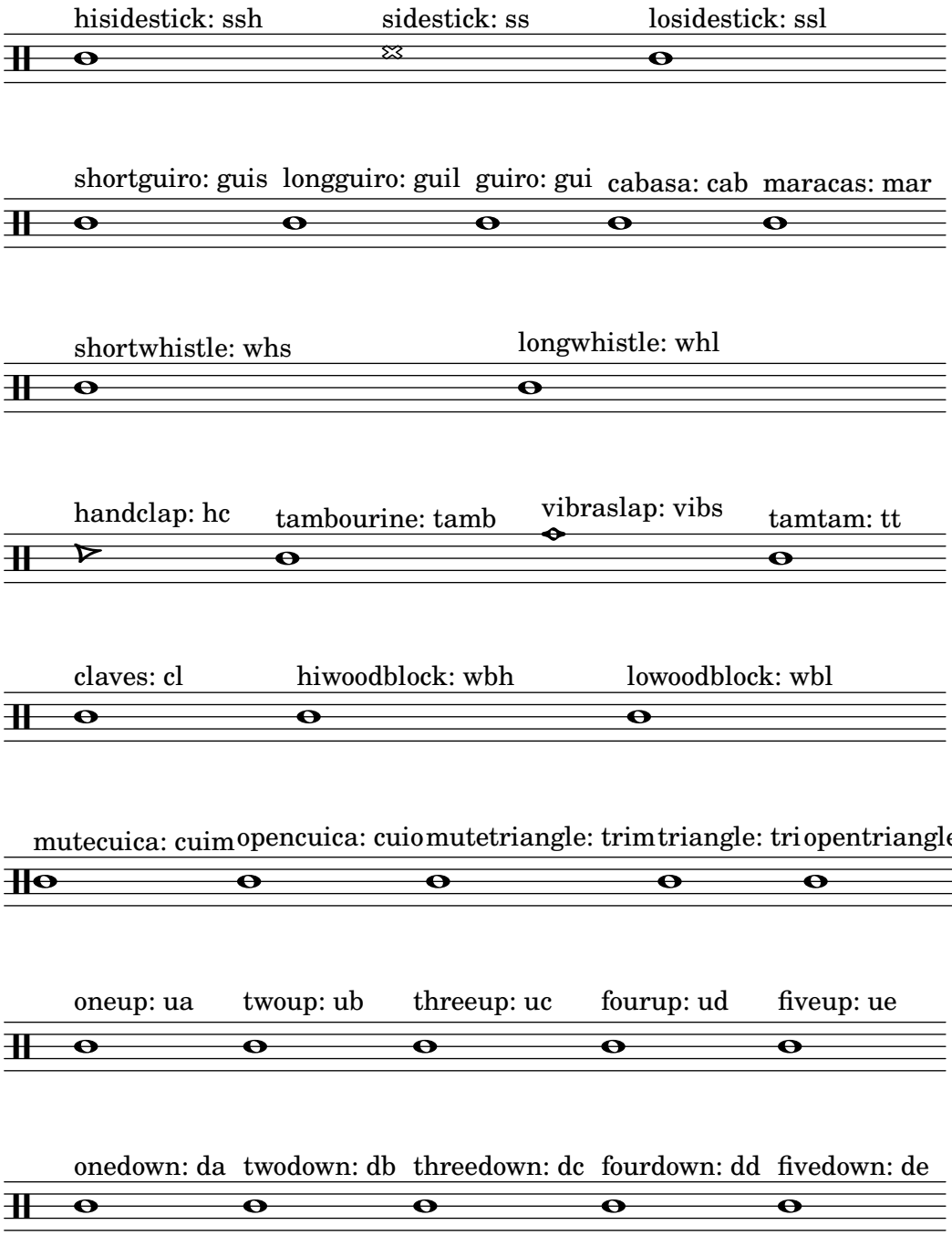
He aquí un cuadro que muestra todas las articulaciones disponibles:





B.11 Notas de percusión





B.12 Todas las propiedades de contexto

`aDueText` (markup)
: Text to print at a unisono passage.

`alignAboveContext` (string)
: Where to insert newly created context in vertiical alignment.

`alignBassFigureAccidentals` (boolean)
: If true, then the accidentals are aligned in bass figure context.

`alignBelowContext` (string)
: Where to insert newly created context in vertiical alignment.

`associatedVoice` (string)
: Name of the `Voice` that has the melody for this `Lyrics` line.

`autoAccidentals` (list)
: List of different ways to typeset an accidental.

  For determining when to print an accidental, several different rules are tried. The rule that gives the highest number of accidentals is used.

  Each entry in the list is either a symbol or a procedure.

  symbol The symbol is the name of the context in which the following rules are to be applied. For example, if *context* is Sección "Score" in *Referencia de Funcionamiento Interno* then all staves share accidentals, and if *context* is Sección "Staff" in *Referencia de Funcionamiento Interno* then all voices in the same staff share accidentals, but staves do not.

  procedure The procedure represents an accidental rule to be applied to the previously specified context.

  The procedure takes the following arguments:

  `context` The current context to which the rule should be applied.

  `pitch` The pitch of the note to be evaluated.

  `barnum` The current bar number.

  `measurepos` The current measure position.

  The procedure returns a pair of booleans. The first states whether an extra natural should be added. The second states whether an accidental should be printed. `(#t . #f)` does not make sense.

`autoBeamCheck` (procedure)
: A procedure taking three arguments, *context*, *dir* [start/stop (-1 or 1)], and *test* [shortest note in the beam]. A non-`#f` return value starts or stops the auto beam.

`autoBeamSettings` (list)
: Specifies when automatically generated beams should begin and end. See Sección "Establecer el comportamiento de las barras automáticas" in *Referencia de la Notación* for more information.

`autoBeaming` (boolean)
: If set to true then beams are generated automatically.

`autoCautionaries` (list)
: List similar to `autoAccidentals`, but it controls cautionary accidentals rather than normal ones. Both lists are tried, and the one giving the most accidentals wins. In case of draw, a normal accidental is typeset.

`automaticBars` (boolean)
: If set to false then bar lines will not be printed automatically; they must be explicitly created with a `\bar` command. Unlike the `\cadenzaOn` keyword, measures are still counted. Bar line generation will resume according to that count if this property is unset.

`barAlways` (boolean)
: If set to true a bar line is drawn after each note.

`barCheckSynchronize` (boolean)
: If true then reset `measurePosition` when finding a bar check.

`barNumberVisibility` (procedure)
: A Procedure that takes an integer and returns whether the corresponding bar number should be printed.

`bassFigureFormatFunction` (procedure)
: A procedure that is called to produce the formatting for a `BassFigure` grob. It takes a list of `BassFigureEvent`s, a context, and the grob to format.

`bassStaffProperties` (list)
: An alist of property settings to apply for the down staff of `PianoStaff`. Used by `\autochange`.

`beatGrouping` (list)
: A list of beatgroups, e.g., in 5/8 time `'(2 3)`.

`beatLength` (moment)
: The length of one beat in this time signature.

`chordChanges` (boolean)
: Only show changes in chords scheme?

`chordNameExceptions` (list)
: An alist of chord exceptions. Contains `(`*`chord . markup`*`)` entries.

`chordNameExceptionsFull` (list)
: An alist of full chord exceptions. Contains `(`*`chord . markup`*`)` entries.

`chordNameExceptionsPartial` (list)
: An alist of partial chord exceptions. Contains `(`*`chord . (prefix-markup suffix-markup`*`))` entries.

`chordNameFunction` (procedure)
: The function that converts lists of pitches to chord names.

`chordNameSeparator` (markup)
: The markup object used to separate parts of a chord name.

`chordNoteNamer` (procedure)
: A function that converts from a pitch object to a text markup. Used for single pitches.

`chordPrefixSpacer` (number)
: The space added between the root symbol and the prefix of a chord name.

`chordRootNamer` (procedure)
: A function that converts from a pitch object to a text markup. Used for chords.

`clefGlyph` (string)
: Name of the symbol within the music font.

`clefOctavation` (integer)
: Add this much extra octavation. Values of 7 and -7 are common.

`clefPosition` (number)
: Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

`completionBusy` (boolean)
: Whether a completion-note head is playing.

`connectArpeggios` (boolean)
: If set, connect arpeggios across piano staff.

`countPercentRepeats` (boolean)
: If set, produce counters for percent repeats.

`createKeyOnClefChange` (boolean)
: Print a key signature whenever the clef is changed.

`createSpacing` (boolean)
: Create `StaffSpacing` objects? Should be set for staves.

`crescendoSpanner` (symbol)
: The type of spanner to be used for crescendi. Available values are '`hairpin`' and '`text`'. If unset, a hairpin crescendo is used.

`crescendoText` (markup)
: The text to print at start of non-hairpin crescendo, i.e., '`cresc.`'.

`currentBarNumber` (integer)
: Contains the current barnumber. This property is incremented at every bar line.

`decrescendoSpanner` (symbol)
: The type of spanner to be used for decrescendi. Available values are '`hairpin`' and '`text`'. If unset, a hairpin decrescendo is used.

`decrescendoText` (markup)
: The text to print at start of non-hairpin decrescendo, i.e., '`dim.`'.

`defaultBarType` (string)
: Set the default type of bar line. See `whichBar` for information on available bar types.

  This variable is read by Sección "Timing_translator" in *Referencia de Funcionamiento Interno* at Sección "Score" in *Referencia de Funcionamiento Interno* level.

`doubleRepeatType` (string)
: Set the default bar line for double repeats.

`doubleSlurs` (boolean)
: If set, two slurs are created for every slurred note, one above and one below the chord.

`drumPitchTable` (hash table)
: A table mapping percussion instruments (symbols) to pitches.

`drumStyleTable` (hash table)
: A hash table which maps drums to layout settings. Predefined values: '`drums-style`', '`timbales-style`', '`congas-style`', '`bongos-style`', and '`percussion-style`'.

  The layout style is a hash table, containing the drum-pitches (e.g., the symbol '`hihat`') as keys, and a list `(`*`notehead-style script vertical-position`*`)` as values.

`explicitClefVisibility` (vector)
: '`break-visibility`' function for clef changes.

`explicitKeySignatureVisibility` (vector)
: '`break-visibility`' function for explicit key changes. '`\override`' of the `break-visibility` property will set the visibility for normal (i.e., at the start of the line) key signatures.

`extendersOverRests` (boolean)
: Whether to continue extenders as they cross a rest.

`extraNatural` (boolean)
: Whether to typeset an extra natural sign before accidentals changing from a non-natural to another non-natural.

`figuredBassAlterationDirection` (direction)
: Where to put alterations relative to the main figure.

`figuredBassCenterContinuations` (boolean)
: Whether to vertically center pairs of extender lines. This does not work with three or more lines.

`figuredBassFormatter` (procedure)
: A routine generating a markup for a bass figure.

`figuredBassPlusDirection` (direction)
: Where to put plus signs relative to the main figure.

`fingeringOrientations` (list)
: A list of symbols, containing '`left`', '`right`', '`up`' and/or '`down`'. This list determines where fingerings are put relative to the chord being fingered.

`firstClef` (boolean)
: If true, create a new clef when starting a staff.

`followVoice` (boolean)
: If set, note heads are tracked across staff switches by a thin line.

`fontSize` (number)
: The relative size of all grobs in a context.

`forbidBreak` (boolean)
: If set to `##t`, prevent a line break at this point.

`forceClef` (boolean)
: Show clef symbol, even if it has not changed. Only active for the first clef after the property is set, not for the full staff.

`gridInterval` (moment)
: Interval for which to generate `GridPoint`s.

`harmonicAccidentals` (boolean)
: If set, harmonic notes in chords get accidentals.

`harmonicDots` (boolean)
: If set, harmonic notes in dotted chords get dots.

`highStringOne` (boolean)
: Whether the first string is the string with highest pitch on the instrument. This used by the automatic string selector for tablature notation.

`ignoreBarChecks` (boolean)
: Ignore bar checks.

`ignoreFiguredBassRest` (boolean)
: Don't swallow rest events.

`ignoreMelismata` (boolean)
: Ignore melismata for this Sección "Lyrics" in *Referencia de Funcionamiento Interno* line.

`implicitBassFigures` (list)
: A list of bass figures that are not printed as numbers, but only as extender lines.

`implicitTimeSignatureVisibility` (vector)
: break visibility for the default time signature.

`instrumentCueName` (markup)
: The name to print if another instrument is to be taken.

`instrumentEqualizer` (procedure)
: A function taking a string (instrument name), and returning a `(min . max)` pair of numbers for the loudness range of the instrument.

`instrumentName` (markup)
: The name to print left of a staff. The `instrument` property labels the staff in the first system, and the `instr` property labels following lines.

`instrumentTransposition` (pitch)
: Define the transposition of the instrument. Its value is the pitch that sounds like middle C. This is used to transpose the MIDI output, and `\quote`s.

`internalBarNumber` (integer)
: Contains the current barnumber. This property is used for internal timekeeping, among others by the `Accidental_engraver`.

`keepAliveInterfaces` (list)
: A list of symbols, signifying grob interfaces that are worth keeping a staff with `remove-empty` set around for.

`keyAlterationOrder` (list)
: An alist that defines in what order alterations should be printed. The format is `(step . alter)`, where *step* is a number from 0 to 6 and *alter* from -2 (sharp) to 2 (flat).

`keySignature` (list)
: The current key signature. This is an alist containing `(step . alter)` or `((octave . step) . alter)`, where *step* is a number in the range 0 to 6 and *alter* a fraction, denoting alteration. For alterations, use symbols, e.g. `keySignature = #`((6 . ,FLAT))`.

`lyricMelismaAlignment` (direction)
: Alignment to use for a melisma syllable.

`majorSevenSymbol` (markup)
: How should the major 7th be formatted in a chord name?

`markFormatter` (procedure)
: A procedure taking as arguments the context and the rehearsal mark. It should return the formatted mark as a markup object.

`maximumFretStretch` (number)
: Don't allocate frets further than this from specified frets.

`measureLength` (moment)
: Length of one measure in the current time signature.

`measurePosition` (moment)
: How much of the current measure have we had. This can be set manually to create incomplete measures.

`melismaBusyProperties` (list)
: A list of properties (symbols) to determine whether a melisma is playing. Setting this property will influence how lyrics are aligned to notes. For example, if set to `#'(melismaBusy beamMelismaBusy)`, only manual melismata and manual beams are considered. Possible values include `melismaBusy`, `slurMelismaBusy`, `tieMelismaBusy`, and `beamMelismaBusy`.

`metronomeMarkFormatter` (procedure)
: How to produce a metronome markup. Called with four arguments: text, duration, count and context.

`middleCClefPosition` (number)
: The position of the middle C, as determined only by the clef. This can be calculated by looking at `clefPosition` and `clefGlyph`.

`middleCOffset` (number)
: The offset of middle C from the position given by `middleCClefPosition` This is used for ottava brackets.

`middleCPosition` (number)
: The place of the middle C, measured in half staff-spaces. Usually determined by looking at `middleCClefPosition` and `middleCOffset`.

`midiInstrument` (string)
: Name of the MIDI instrument to use.

`midiMaximumVolume` (number)
: Analogous to `midiMinimumVolume`.

`midiMinimumVolume` (number)
: Set the minimum loudness for MIDI. Ranges from 0 to 1.

`minimumFret` (number)
: The tablature auto string-selecting mechanism selects the highest string with a fret at least `minimumFret`.

`minimumPageTurnLength` (moment)
: Minimum length of a rest for a page turn to be allowed.

`minimumRepeatLengthForPageTurn` (moment)
: Minimum length of a repeated section for a page turn to be allowed within that section.

`noteToFretFunction` (procedure)
: How to produce a fret diagram. Parameters: A list of note events and a list of tabstring events.

`ottavation` (markup)
: If set, the text for an ottava spanner. Changing this creates a new text spanner.

`output` (unknown)
: The output produced by a score-level translator during music interpretation.

`pedalSostenutoStrings` (list)
: See `pedalSustainStrings`.

`pedalSostenutoStyle` (symbol)
: See `pedalSustainStyle`.

`pedalSustainStrings` (list)
: A list of strings to print for sustain-pedal. Format is `(up updown down)`, where each of the three is the string to print when this is done with the pedal.

`pedalSustainStyle` (symbol)
: A symbol that indicates how to print sustain pedals: `text`, `bracket` or `mixed` (both).

`pedalUnaCordaStrings` (list)
: See `pedalSustainStrings`.

`pedalUnaCordaStyle` (symbol)
: See `pedalSustainStyle`.

`predefinedDiagramTable` (hash table)
: The hash table of predefined fret diagrams to use in FretBoards.

`printKeyCancellation` (boolean)
: Print restoration alterations before a key signature change.

`printOctaveNames` (boolean)
: Print octave marks for the `NoteNames` context.

`printPartCombineTexts` (boolean)
: Set 'Solo' and 'A due' texts in the part combiner?

`proportionalNotationDuration` (moment)
: Global override for shortest-playing duration. This is used for switching on proportional notation.

`recordEventSequence` (procedure)
: When `Recording_group_engraver` is in this context, then upon termination of the context, this function is called with current context and a list of music objects. The list of contains entries with start times, music objects and whether they are processed in this context.

`rehearsalMark` (integer)
: The last rehearsal mark printed.

`repeatCommands` (list)
: This property is a list of commands of the form `(list 'volta` *x*`)`, where *x* is a string or `#f`. `'end-repeat` is also accepted as a command.

`repeatCountVisibility` (procedure)
: A procedure taking as arguments an integer and context, returning whether the corresponding percent repeat number should be printed when `countPercentRepeats` is set.

`restNumberThreshold` (number)
: If a multimeasure rest has more measures than this, a number is printed.

`shapeNoteStyles` (vector)
: Vector of symbols, listing style for each note head relative to the tonic (qv.) of the scale.

`shortInstrumentName` (markup)
: See `instrument`.

`shortVocalName` (markup)
: Name of a vocal line, short version.

`skipBars` (boolean)
: If set to true, then skip the empty bars that are produced by multimeasure notes and rests. These bars will not appear on the printed output. If not set (the default), multimeasure notes and rests expand into their full length, printing the appropriate number of empty bars so that synchronization with other voices is preserved.

```
{
  r1 r1*3 R1*3
  \set Score.skipBars= ##t
  r1*3 R1*3
}
```

`skipTypesetting` (boolean)
: If true, no typesetting is done, speeding up the interpretation phase. Useful for debugging large scores.

`soloIIText` (markup)
: The text for the start of a solo for voice 'two' when part-combining.

`soloText` (markup)
: The text for the start of a solo when part-combining.

`squashedPosition` (integer)
: Vertical position of squashing for Sección "Pitch_squash_engraver" in *Referencia de Funcionamiento Interno*.

`staffLineLayoutFunction` (procedure)
: Layout of staff lines, `traditional`, or `semitone`.

`stanza` (markup)
: Stanza 'number' to print before the start of a verse. Use in `Lyrics` context.

`stemLeftBeamCount` (integer)
: Specify the number of beams to draw on the left side of the next note. Overrides automatic beaming. The value is only used once, and then it is erased.

`stemRightBeamCount` (integer)
: See `stemLeftBeamCount`.

`stringNumberOrientations` (list)
: See `fingeringOrientations`.

`stringOneTopmost` (boolean)
: Whether the first string is printed on the top line of the tablature.

`stringTunings` (list)
: The tablature strings tuning. It is a list of the pitch (in semitones) of each string (starting with the lower one).

`strokeFingerOrientations` (list)
: See `fingeringOrientations`.

`subdivideBeams` (boolean)
: If set, multiple beams will be subdivided at beat positions by only drawing one beam over the beat.

`suggestAccidentals` (boolean)
: If set, accidentals are typeset as cautionary suggestions over the note.

`systemStartDelimiter` (symbol)
: Which grob to make for the start of the system/staff? Set to `SystemStartBrace`, `SystemStartBracket` or `SystemStartBar`.

`systemStartDelimiterHierarchy` (pair)
: A nested list, indicating the nesting of a start delimiters.

`tablatureFormat` (procedure)
: A function formatting a tablature note head. Called with three arguments: string number, context and event. It returns the text as a string.

`tempoHideNote` (boolean)
: Hide the note=count in tempo marks.

`tempoText` (markup)
: Text for tempo marks.

`tempoUnitCount` (number)
: Count for specifying tempo.

`tempoUnitDuration` (duration)
: Unit for specifying tempo.

`tempoWholesPerMinute` (moment)
: The tempo in whole notes per minute.

`tieWaitForNote` (boolean)
: If true, tied notes do not have to follow each other directly. This can be used for writing out arpeggios.

`timeSignatureFraction` (pair of numbers)
: A pair of numbers, signifying the time signature. For example, `#'(4 . 4)` is a 4/4 time signature.

`timing` (boolean)
: Keep administration of measure length, position, bar number, etc.? Switch off for cadenzas.

`tonic` (pitch)
: The tonic of the current scale.

`trebleStaffProperties` (list)
: An alist of property settings to apply for the up staff of `PianoStaff`. Used by `\autochange`.

`tremoloFlags` (integer)
: The number of tremolo flags to add if no number is specified.

`tupletFullLength` (boolean)
: If set, the tuplet is printed up to the start of the next note.

`tupletFullLengthNote` (boolean)
: If set, end at the next note, otherwise end on the matter (time signatures, etc.) before the note.

`tupletSpannerDuration` (moment)
: Normally, a tuplet bracket is as wide as the `\times` expression that gave rise to it. By setting this property, you can make brackets last shorter.

```
{
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```

`useBassFigureExtenders` (boolean)
: Whether to use extender lines for repeated bass figures.

`verticallySpacedContexts` (list)
: List of symbols, containing context names whose vertical axis groups should be taken into account for vertical spacing of systems.

`vocalName` (markup)
: Name of a vocal line.

`voltaSpannerDuration` (moment)
: This specifies the maximum duration to use for the brackets printed for `\alternative`. This can be used to shrink the length of brackets in the situation where one alternative is very large.

`whichBar` (string)
: This property is read to determine what type of bar line to create.

  Example:

  ```
  \set Staff.whichBar = "|:"
  ```

  This will create a start-repeat bar in this staff only. Valid values are described in Sección “bar-line-interface” in *Referencia de Funcionamiento Interno*.

B.13 Propiedades de disposición

`X-extent` (pair of numbers)
: Hard coded extent in X direction.

`X-offset` (number)
: The horizontal amount that this object is moved relative to its X-parent.

`Y-extent` (pair of numbers)
: Hard coded extent in Y direction.

`Y-offset` (number)
: The vertical amount that this object is moved relative to its Y-parent.

`add-stem-support` (boolean)
: If set, the `Stem` object is included in this script's support.

`after-line-breaking` (boolean)
: Dummy property, used to trigger callback for `after-line-breaking`.

`align-dir` (direction)
: Which side to align? `-1`: left side, `0`: around center of width, `1`: right side.

`allow-loose-spacing` (boolean)
: If set, column can be detached from main spacing.

`allow-span-bar` (boolean)
: If false, no inter-staff bar line will be created below this bar line.

`alteration` (number)
: Alteration numbers for accidental.

`alteration-alist` (list)
: List of `(`*`pitch . accidental`*`)` pairs for key signature.

`annotation` (string)
: Annotate a grob for debug purposes.

`arpeggio-direction` (direction)
: If set, put an arrow on the arpeggio squiggly line.

`arrow-length` (number)
: Arrow length.

`arrow-width` (number)
: Arrow width.

`auto-knee-gap` (dimension, in staff space)
: If a gap is found between note heads where a horizontal beam fits that is larger than this number, make a kneed beam.

`average-spacing-wishes` (boolean)
: If set, the spacing wishes are averaged over staves.

`avoid-note-head` (boolean)
: If set, the stem of a chord does not pass through all note heads, but starts at the last note head.

`avoid-slur` (symbol)
: Method of handling slur collisions. Choices are `around`, `inside`, `outside`. If unset, scripts and slurs ignore each other. `around` only moves the script if there is a collision; `outside` always moves the script.

`axes` (list)
: List of axis numbers. In the case of alignment grobs, this should contain only one number.

`bar-size` (dimension, in staff space)
: The size of a bar line.

`base-shortest-duration` (moment)
: Spacing is based on the shortest notes in a piece. Normally, pieces are spaced as if notes at least as short as this are present.

`baseline-skip` (dimension, in staff space)
: Distance between base lines of multiple lines of text.

`beam-thickness` (dimension, in staff space)
: Beam thickness, measured in `staff-space` units.

`beam-width` (dimension, in staff space)
: Width of the tremolo sign.

`beamed-stem-shorten` (list)
: How much to shorten beamed stems, when their direction is forced. It is a list, since the value is different depending on the number of flags and beams.

`beaming` (pair)
: Pair of number lists. Each number list specifies which beams to make. `0` is the central beam, `1` is the next beam toward the note, etc. This information is used to determine how to connect the beaming patterns from stem to stem inside a beam.

`before-line-breaking` (boolean)
: Dummy property, used to trigger a callback function.

`between-cols` (pair)
: Where to attach a loose column to.

`bound-details` (list)
: An alist of properties for determining attachments of spanners to edges.

`bound-padding` (number)
: The amount of padding to insert around spanner bounds.

`bracket-flare` (pair of numbers)
: A pair of numbers specifying how much edges of brackets should slant outward. Value `0.0` means straight edges.

`bracket-visibility` (boolean or symbol)
: This controls the visibility of the tuplet bracket. Setting it to false prevents printing of the bracket. Setting the property to `if-no-beam` makes it print only if there is no beam associated with this tuplet bracket.

`break-align-anchor` (number)
: Grobs aligned to this break-align grob will have their X-offsets shifted by this number. In bar lines, for example, this is used to position grobs relative to the (visual) center of the bar line.

`break-align-anchor-alignment` (number)
: Read by `ly:break-aligned-interface::calc-extent-aligned-anchor` for aligning an anchor to a grob's extent

`break-align-orders` (vector)
: Defines the order in which prefatory matter (clefs, key signatures) appears. The format is a vector of length 3, where each element is one order for end-of-line, middle of line, and start-of-line, respectively. An order is a list of symbols.

  For example, clefs are put after key signatures by setting

```
\override Score.BreakAlignment #'break-align-orders =
  #(make-vector 3 '(span-bar
                    breathing-sign
                    staff-bar
                    key
                    clef
                    time-signature))
```

`break-align-symbol` (symbol)
: This key is used for aligning and spacing breakable items.

`break-align-symbols` (list)
: A list of symbols that determine which break-aligned grobs to align this to. If the grob selected by the first symbol in the list is invisible due to break-visibility, we will align to the next grob (and so on).

`break-overshoot` (pair of numbers)
: How much does a broken spanner stick out of its bounds?

`break-visibility` (vector)
: A vector of 3 booleans, `#(`*`end-of-line unbroken begin-of-line`*`)`. `#t` means visible, `#f` means killed.

`breakable` (boolean)
: Allow breaks here.

`c0-position` (integer)
: An integer indicating the position of middle C.

`circled-tip` (boolean)
: Put a circle at start/end of hairpins (al/del niente).

`clip-edges` (boolean)
: Allow outward pointing beamlets at the edges of beams?

`collapse-height` (dimension, in staff space)
: Minimum height of system start delimiter. If equal or smaller, the bracket/brace/line is removed.

`color` (list)
: The color of this grob.

`common-shortest-duration` (moment)
: The most common shortest note length. This is used in spacing. Enlarging this sets the score tighter.

`concaveness` (number)
: A beam is concave if its inner stems are closer to the beam than the two outside stems. This number is a measure of the closeness of the inner stems. It is used for damping the slope of the beam.

`connect-to-neighbor` (pair)
: Pair of booleans, indicating whether this grob looks as a continued break.

`control-points` (list)
: List of offsets (number pairs) that form control points for the tie, slur, or bracket shape. For Béziers, this should list the control points of a third-order Bézier curve.

`damping` (number)
: Amount of beam slope damping.

`dash-fraction` (number)
: Size of the dashes, relative to `dash-period`. Should be between `0.0` (no line) and `1.0` (continuous line).

`dash-period` (number)
: The length of one dash together with whitespace. If negative, no line is drawn at all.

`default-direction` (direction)
: Direction determined by note head positions.

`digit-names` (unknown)
: Names for string finger digits.

`direction` (direction)
: If `side-axis` is `0` (or `#X`), then this property determines whether the object is placed `#LEFT`, `#CENTER` or `#RIGHT` with respect to the other object. Otherwise, it determines whether the object is placed `#UP`, `#CENTER` or `#DOWN`. Numerical values may also be used: `#UP=1`, `#DOWN=-1`, `#LEFT=-1`, `#RIGHT=1`, `#CENTER=0`.

`dot-count` (integer)
: The number of dots.

`dot-negative-kern` (number)
: The space to remove between a dot and a slash in percent repeat glyphs. Larger values bring the two elements closer together.

`dot-placement-list` (list)
: List consisting of `(`*`description string-number fret-number finger-number`*`)` entries used to define fret diagrams.

`duration-log` (integer)
: The 2-log of the note head duration, i.e., `0` = whole note, `1` = half note, etc.

`eccentricity` (number)
: How asymmetrical to make a slur. Positive means move the center to the right.

`edge-height` (pair)
: A pair of numbers specifying the heights of the vertical edges: `(`*`left-height . right-height`*`)`.

`edge-text` (pair)
: A pair specifying the texts to be set at the edges: `(`*`left-text . right-text`*`)`.

`expand-limit` (integer)
: Maximum number of measures expanded in church rests.

`extra-X-extent` (pair of numbers)
: A grob is enlarged in X dimension by this much.

`extra-Y-extent` (pair of numbers)
: A grob is enlarged in Y dimension by this much.

`extra-dy` (number)
: Slope glissandi this much extra.

`extra-offset` (pair of numbers)
: A pair representing an offset. This offset is added just before outputting the symbol, so the typesetting engine is completely oblivious to it. The values are measured in `staff-space` units of the staff's `StaffSymbol`.

`extra-spacing-height` (pair of numbers)
: In the horizontal spacing problem, we increase the height of each item by this amount (by adding the 'car' to the bottom of the item and adding the 'cdr' to the top of the item. In order to make a grob infinitely high (to prevent the horizontal spacing problem from placing any other grobs above or below this grob), set this to `(-inf.0 . +inf.0)`.

`extra-spacing-width` (pair of numbers)
: In the horizontal spacing problem, we pad each item by this amount (by adding the 'car' on the left side of the item and adding the 'cdr' on the right side of the item). In order to make a grob take up no horizontal space at all, set this to `(+inf.0 . -inf.0)`.

`flag` (unknown)
: A function returning the full flag stencil for the `Stem`, which is passed to the function as the only argument. The default ly:stem::calc-stencil function uses the `flag-style` property to determine the correct glyph for the flag. By providing your own function, you can create arbitrary flags.

`flag-count` (number)
: The number of tremolo beams.

`flag-style` (symbol)
: A symbol determining what style of flag glyph is typeset on a `Stem`. Valid options include `'()` for standard flags, `'mensural` and `'no-flag`, which switches off the flag.

`font-encoding` (symbol)
: The font encoding is the broadest category for selecting a font. Currently, only lilypond's system fonts (Emmentaler and Aybabtu) are using this property. Available values are `fetaMusic` (Emmentaler), `fetaBraces` (Aybabtu), `fetaNumber` (Emmentaler), and `fetaDynamic` (Emmentaler).

`font-family` (symbol)
: The font family is the broadest category for selecting text fonts. Options include: `sans`, `roman`.

`font-name` (string)
: Specifies a file name (without extension) of the font to load. This setting overrides selection using `font-family`, `font-series` and `font-shape`.

`font-series` (symbol)
: Select the series of a font. Choices include `medium`, `bold`, `bold-narrow`, etc.

`font-shape` (symbol)
: Select the shape of a font. Choices include `upright`, `italic`, `caps`.

`font-size` (number)
: The font size, compared to the 'normal' size. `0` is style-sheet's normal size, `-1` is smaller, `+1` is bigger. Each step of 1 is approximately 12% larger; 6 steps are exactly a factor 2 larger. Fractional values are allowed.

`force-hshift` (number)
: This specifies a manual shift for notes in collisions. The unit is the note head width of the first voice note. This is used by Sección “note-collision-interface” in *Referencia de Funcionamiento Interno*.

`fraction` (pair of numbers)
: Numerator and denominator of a time signature object.

`french-beaming` (boolean)
: Use French beaming style for this stem. The stem stops at the innermost beams.

`fret-diagram-details` (list)
: An alist of detailed grob properties for fret diagrams. Each alist entry consists of a (`property . value`) pair. The properties which can be included in fret-diagram-details include the following:

  - `barre-type` – Type of barre indication used. Choices include `curved`, `straight`, and `none`. Default `curved`.
  - `capo-thickness` – Thickness of capo indicator, in multiples of fret-space. Default value 0.5.
  - `dot-color` – Color of dots. Options include `black` and `white`. Default `black`.
  - `dot-label-font-mag` – Magnification for font used to label fret dots. Default value 1.
  - `dot-radius` – Radius of dots, in terms of fret spaces. Default value 0.425 for labeled dots, 0.25 for unlabeled dots.
  - `finger-code` – Code for the type of fingering indication used. Options include `none`, `in-dot`, and `below-string`. Default `none` for markup fret diagrams, `below-string` for `FretBoards` fret diagrams.
  - `fret-count` – The number of frets. Default 4.
  - `fret-label-font-mag` – The magnification of the font used to label the lowest fret number. Default 0.5
  - `fret-label-vertical-offset` – The vertical offset of the fret label from the fret. Default -0.2
  - `label-dir` – Side to which the fret label is attached. `-1`, `#LEFT`, or `#DOWN` for left or down; `1`, `#RIGHT`, or `#UP` for right or up. Default `#RIGHT`.
  - `mute-string` – Character string to be used to indicate muted string. Default `"x"`.
  - `number-type` – Type of numbers to use in fret label. Choices include `roman-lower`, `roman-upper`, and `arabic`. Default `roman-lower`.
  - `open-string` – Character string to be used to indicate open string. Default `"o"`.
  - `orientation` – Orientation of fret-diagram. Options include `normal` and `landscape`. Default `normal`.
  - `string-count` – The number of strings. Default 6.
  - `string-label-font-mag` – The magnification of the font used to label fingerings at the string, rather than in the dot. Default value 0.6.
  - `top-fret-thickness` – The thickness of the top fret line, as a multiple of the standard thickness. Default value 3.
  - `xo-font-magnification` – Magnification used for mute and open string indicators. Default value 0.5.

- `xo-padding` – Padding for open and mute indicators from top fret. Default value 0.25.

`full-length-padding` (number)
: How much padding to use at the right side of a full-length tuplet bracket.

`full-length-to-extent` (boolean)
: Run to the extent of the column for a full-length tuplet bracket.

`full-size-change` (boolean)
: Don't make a change clef smaller.

`gap` (dimension, in staff space)
: Size of a gap in a variable symbol.

`gap-count` (integer)
: Number of gapped beams for tremolo.

`glyph` (string)
: A string determining what 'style' of glyph is typeset. Valid choices depend on the function that is reading this property.

`glyph-name-alist` (list)
: An alist of key-string pairs.

`grow-direction` (direction)
: Crescendo or decrescendo?

`hair-thickness` (number)
: Thickness of the thin line in a bar line.

`harp-pedal-details` (list)
: An alist of detailed grob properties for harp pedal diagrams. Each alist entry consists of a (`property` . `value`) pair. The properties which can be included in harp-pedal-details include the following:

  - `box-offset` – Vertical shift of the center of flat / sharp pedal boxes above / below the horizontal line. Default value 0.8.
  - `box-width` – Width of each pedal box. Default value 0.4.
  - `box-height` – Height of each pedal box. Default value 1.0.
  - `space-before-divider` – Space between boxes before the first divider (so that the diagram can be made symmetric). Default value 0.8.
  - `space-after-divider` – Space between boxes after the first divider. Default value 0.8.
  - `circle-thickness` – Thickness (in unit of the line-thickness) of the ellipse around circled pedals. Default value 0.5.
  - `circle-x-padding` – Padding in X direction of the ellipse around circled pedals. Default value 0.15.
  - `circle-y-padding` – Padding in Y direction of the ellipse around circled pedals. Default value 0.2.

`head-direction` (direction)
: Are the note heads left or right in a semitie?

`height` (dimension, in staff space)
: Height of an object in `staff-space` units.

`height-limit` (dimension, in staff space)
: Maximum slur height: The longer the slur, the closer it is to this height.

`horizontal-shift` (integer)
: An integer that identifies ranking of `NoteColumn`s for horizontal shifting. This is used by Sección “note-collision-interface” in *Referencia de Funcionamiento Interno*.

`horizontal-skylines` (unknown)
: Two skylines, one to the left and one to the right of this grob.

`ignore-collision` (boolean)
: If set, don’t do note collision resolution on this `NoteColumn`.

`implicit` (boolean)
: Is this an implicit bass figure?

`inspect-index` (integer)
: If debugging is set, set beam and slur configuration to this index, and print the respective scores.

`inspect-quants` (pair of numbers)
: If debugging is set, set beam and slur quants to this position, and print the respective scores.

`keep-fixed-while-stretching` (boolean)
: A grob with this property set to true is fixed relative to the staff above it when systems are stretched.

`keep-inside-line` (boolean)
: If set, this column cannot have objects sticking into the margin.

`kern` (dimension, in staff space)
: Amount of extra white space to add. For bar lines, this is the amount of space after a thick line.

`knee` (boolean)
: Is this beam kneed?

`knee-spacing-correction` (number)
: Factor for the optical correction amount for kneed beams. Set between `0` for no correction and `1` for full correction.

`labels` (list)
: List of labels (symbols) placed on a column

`layer` (integer)
: The output layer (a value between 0 and 2: Layers define the order of printing objects. Objects in lower layers are overprinted by objects in higher layers.

`ledger-line-thickness` (pair of numbers)
: The thickness of ledger lines. It is the sum of 2 numbers: The first is the factor for line thickness, and the second for staff space. Both contributions are added.

`left-bound-info` (list)
: An alist of properties for determining attachments of spanners to edges.

`left-padding` (dimension, in staff space)
: The amount of space that is put left to an object (e.g., a group of accidentals).

`length` (dimension, in staff space)
: User override for the stem length of unbeamed stems.

`length-fraction` (number)
: Multiplier for lengths. Used for determining ledger lines and stem lengths.

`line-break-penalty` (number)
: Penalty for a line break at this column. This affects the choices of the line breaker; it avoids a line break at a column with a positive penalty and prefers a line break at a column with a negative penalty.

`line-break-permission` (symbol)
: Instructs the line breaker on whether to put a line break at this column. Can be `force` or `allow`.

`line-break-system-details` (list)
: An alist of properties to use if this column is the start of a system.

`line-count` (integer)
: The number of staff lines.

`line-positions` (list)
: Vertical positions of staff lines.

`line-thickness` (number)
: The thickness of the tie or slur contour.

`long-text` (markup)
: Text markup. See Sección “Formatear el texto” in *Referencia de la Notación*.

`max-beam-connect` (integer)
: Maximum number of beams to connect to beams from this stem. Further beams are typeset as beamlets.

`max-stretch` (number)
: The maximum amount that this `VerticalAxisGroup` can be vertically stretched (for example, in order to better fill a page).

`measure-count` (integer)
: The number of measures for a multi-measure rest.

`measure-length` (moment)
: Length of a measure. Used in some spacing situations.

`merge-differently-dotted` (boolean)
: Merge note heads in collisions, even if they have a different number of dots. This is normal notation for some types of polyphonic music.

  `merge-differently-dotted` only applies to opposing stem directions (i.e., voice 1 & 2).

`merge-differently-headed` (boolean)
: Merge note heads in collisions, even if they have different note heads. The smaller of the two heads is rendered invisible. This is used in polyphonic guitar notation. The value of this setting is used by Sección “note-collision-interface” in *Referencia de Funcionamiento Interno*.

  `merge-differently-headed` only applies to opposing stem directions (i.e., voice 1 & 2).

`minimum-X-extent` (pair of numbers)
: Minimum size of an object in X dimension, measured in `staff-space` units.

`minimum-Y-extent` (pair of numbers)
: Minimum size of an object in Y dimension, measured in `staff-space` units.

`minimum-distance` (dimension, in staff space)
: Minimum distance between rest and notes or beam.

`minimum-length` (dimension, in staff space)
: Try to make a spanner at least this long, normally in the horizontal direction. This requires an appropriate callback for the `springs-and-rods` property. If added to a `Tie`, this sets the minimum distance between noteheads.

`minimum-length-fraction` (number)
: Minimum length of ledger line as fraction of note head size.

`minimum-space` (dimension, in staff space)
: Minimum distance that the victim should move (after padding).

`neutral-direction` (direction)
: Which direction to take in the center of the staff.

`neutral-position` (number)
: Position (in half staff spaces) where to flip the direction of custos stem.

`next` (layout object)
: Object that is next relation (e.g., the lyric syllable following an extender.

`no-alignment` (boolean)
: If set, don't place this grob in a `VerticalAlignment`; rather, place it using its own `Y-offset` callback.

`no-ledgers` (boolean)
: If set, don't draw ledger lines on this object.

`no-stem-extend` (boolean)
: If set, notes with ledger lines do not get stems extending to the middle staff line.

`non-default` (boolean)
: Set for manually specified clefs.

`non-musical` (boolean)
: True if the grob belongs to a `NonMusicalPaperColumn`.

`note-names` (vector)
: Vector of strings containing names for easy-notation note heads.

`outside-staff-horizontal-padding` (number)
: By default, an outside-staff-object can be placed so that is it very close to another grob horizontally. If this property is set, the outside-staff-object is raised so that it is not so close to its neighbor.

`outside-staff-padding` (number)
: The padding to place between this grob and the staff when spacing according to `outside-staff-priority`.

`outside-staff-priority` (number)
: If set, the grob is positioned outside the staff in such a way as to avoid all collisions. In case of a potential collision, the grob with the smaller `outside-staff-priority` is closer to the staff.

`packed-spacing` (boolean)
: If set, the notes are spaced as tightly as possible.

`padding` (dimension, in staff space)
: Add this much extra space between objects that are next to each other.

`padding-pairs` (list)
: An alist mapping `(`*`name . name`*`)` to distances.

`page-break-penalty` (number)
: Penalty for page break at this column. This affects the choices of the page breaker; it avoids a page break at a column with a positive penalty and prefers a page break at a column with a negative penalty.

`page-break-permission` (symbol)
: Instructs the page breaker on whether to put a page break at this column. Can be `force` or `allow`.

`page-turn-penalty` (number)
: Penalty for a page turn at this column. This affects the choices of the page breaker; it avoids a page turn at a column with a positive penalty and prefers a page turn at a column with a negative penalty.

`page-turn-permission` (symbol)
: Instructs the page breaker on whether to put a page turn at this column. Can be `force` or `allow`.

`parenthesized` (boolean)
: Parenthesize this grob.

`positions` (pair of numbers)
: Pair of staff coordinates `(`*`left . right`*`)`, where both *left* and *right* are in `staff-space` units of the current staff. For slurs, this value selects which slur candidate to use; if extreme positions are requested, the closest one is taken.

`prefer-dotted-right` (boolean)
: For note collisions, prefer to shift dotted up-note to the right, rather than shifting just the dot.

`ratio` (number)
: Parameter for slur shape. The higher this number, the quicker the slur attains its `height-limit`.

`remove-empty` (boolean)
: If set, remove group if it contains no interesting items.

`remove-first` (boolean)
: Remove the first staff of an orchestral score?

`restore-first` (boolean)
: Print a natural before the accidental.

`rhythmic-location` (rhythmic location)
: Where (bar number, measure position) in the score.

`right-bound-info` (list)
: An alist of properties for determining attachments of spanners to edges.

`right-padding` (dimension, in staff space)
: Space to insert on the right side of an object (e.g., between note and its accidentals).

`rotation` (list)
: Number of degrees to rotate this object, and what point to rotate around. For example, `#'(45 0 0)` rotates by 45 degrees around the center of this object.

`same-direction-correction` (number)
: Optical correction amount for stems that are placed in tight configurations. This amount is used for stems with the same direction to compensate for note head to stem distance.

`script-priority` (number)
: A sorting key that determines in what order a script is within a stack of scripts.

`self-alignment-X` (number)
: Specify alignment of an object. The value `-1` means left aligned, `0` centered, and `1` right-aligned in X direction. Other numerical values may also be specified.

`self-alignment-Y` (number)
: Like `self-alignment-X` but for the Y axis.

`shorten-pair` (pair of numbers)
: The lengths to shorten a text-spanner on both sides, for example a pedal bracket. Positive values shorten the text-spanner, while negative values lengthen it.

`shortest-duration-space` (dimension, in staff space)
: Start with this much space for the shortest duration. This is expressed in `spacing-increment` as unit. See also Sección "spacing-spanner-interface" in *Referencia de Funcionamiento Interno*.

`shortest-playing-duration` (moment)
: The duration of the shortest note playing here.

`shortest-starter-duration` (moment)
: The duration of the shortest note that starts here.

`side-axis` (number)
: If the value is `#X` (or equivalently `0`), the object is placed horizontally next to the other object. If the value is `#Y` or `1`, it is placed vertically.

`side-relative-direction` (direction)
: Multiply direction of `direction-source` with this to get the direction of this object.

`size` (number)
: Size of object, relative to standard size.

`skyline-horizontal-padding` (number)
: For determining the vertical distance between two staves, it is possible to have a configuration which would result in a tight interleaving of grobs from the top staff and the bottom staff. The larger this parameter is, the farther apart the staves are placed in such a configuration.

`slash-negative-kern` (number)
: The space to remove between slashes in percent repeat glyphs. Larger values bring the two elements closer together.

`slope` (number)
: The slope of this object.

`slur-padding` (number)
: Extra distance between slur and script.

`space-alist` (list)
: A table that specifies distances between prefatory items, like clef and time-signature. The format is an alist of spacing tuples: `(`*`break-align-symbol type . distance`*`)`, where *type* can be the symbols `minimum-space` or `extra-space`.

`space-to-barline` (boolean)
: If set, the distance between a note and the following non-musical column will be measured to the bar line instead of to the beginning of the non-musical column. If there is a clef change followed by a bar line, for example, this means that we will try to space the non-musical column as though the clef is not there.

`spacing-increment` (number)
: Add this much space for a doubled duration. Typically, the width of a note head. See also Sección "spacing-spanner-interface" in *Referencia de Funcionamiento Interno*.

`springs-and-rods` (boolean)
: Dummy variable for triggering spacing routines.

`stacking-dir` (direction)
: Stack objects in which direction?

`staff-padding` (dimension, in staff space)
: Maintain this much space between reference points and the staff. Its effect is to align objects of differing sizes (like the dynamics **p** and **f**) on their baselines.

`staff-position` (number)
: Vertical position, measured in half staff spaces, counted from the middle line.

`staff-space` (dimension, in staff space)
: Amount of space between staff lines, expressed in global `staff-space`.

`stem-attachment` (pair of numbers)
: An `(x . y)` pair where the stem attaches to the notehead.

`stem-end-position` (number)
: Where does the stem end (the end is opposite to the support-head)?

`stem-spacing-correction` (number)
: Optical correction amount for stems that are placed in tight configurations. For opposite directions, this amount is the correction for two normal sized stems that overlap completely.

`stemlet-length` (number)
: How long should a stem over a rest be?

`stencil` (unknown)
: The symbol to print.

`stencils` (list)
: Multiple stencils, used as intermediate value.

`strict-grace-spacing` (boolean)
: If set, main notes are spaced normally, then grace notes are put left of the musical columns fot the main notes.

`strict-note-spacing` (boolean)
: If set, unbroken columns with non-musical material (clefs, bar lines, etc.) are not spaced separately, but put before musical columns.

`stroke-style` (string)
: Set to `"grace"` to turn stroke through flag on.

`style` (symbol)
: This setting determines in what style a grob is typeset. Valid choices depend on the `stencil` callback reading this property.

`text` (markup)
: Text markup. See Sección "Formatear el texto" in *Referencia de la Notación*.

`text-direction` (direction)
: This controls the ordering of the words. The default `RIGHT` is for roman text. Arabic or Hebrew should use `LEFT`.

`thick-thickness` (number)
: Bar line thickness, measured in `line-thickness`.

`thickness` (number)
: Line thickness, generally measured in `line-thickness`.

`thin-kern` (number)
: The space after a hair-line in a bar line.

`threshold` (pair of numbers)
: `(min . max)`, where *min* and *max* are dimensions in staff space.

`tie-configuration` (list)
: List of `(position . dir)` pairs, indicating the desired tie configuration, where *position* is the offset from the center of the staff in staff space and *dir* indicates the direction of the tie (`1`=>up, `-1`=>down, `0`=>center). A non-pair entry in the list causes the corresponding tie to be formatted automatically.

`to-barline` (boolean)
: If true, the spanner will stop at the bar line just before it would otherwise stop.

`toward-stem-shift` (number)
: Amount by which scripts are shifted toward the stem if their direction coincides with the stem direction. `0.0` means keep the default position (centered on the note head), `1.0` means centered on the stem. Interpolated values are possible.

`transparent` (boolean)
: This makes the grob invisible.

`uniform-stretching` (boolean)
: If set, items stretch proportionally to their durations. This looks better in complex polyphonic patterns.

`used` (boolean)
: If set, this spacing column is kept in the spacing problem.

`vertical-skylines` (unknown)
: Two skylines, one above and one below this grob.

`when` (moment)
: Global time step associated with this column happen?

`width` (dimension, in staff space)
: The width of a grob measured in staff space.

`word-space` (dimension, in staff space)
: Space to insert between words in texts.

`zigzag-length` (dimension, in staff space)
: The length of the lines of a zigzag, relative to `zigzag-width`. A value of `1` gives 60-degree zigzags.

`zigzag-width` (dimension, in staff space)
: The width of one zigzag squiggle. This number is adjusted slightly so that the glissando line can be constructed from a whole number of squiggles.

B.14 Identificadores

`acciaccatura` - *music* (music)
: (undocumented; fixme)

`addChordShape` - *key-symbol* (symbol) *tuning* (pair) *shape-definition* (unknown)
: Add chord shape `shape-definition` to the `chord-shape-table` hash with the key `(cons key-symbol tuning)`.

`addInstrumentDefinition` - *name* (string) *lst* (list)
: (undocumented; fixme)

`addQuote` - *name* (string) *music* (music)
: Add a piece of music to be quoted

`afterGrace` - *main* (music) *grace* (music)
: (undocumented; fixme)

`allowPageTurn`
: Allow a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

`applyContext` - *proc* (procedure)
: (undocumented; fixme)

`applyMusic` - *func* (procedure) *music* (music)
: (undocumented; fixme)

`applyOutput` - *ctx* (symbol) *proc* (procedure)
: (undocumented; fixme)

`appoggiatura` - *music* (music)
: (undocumented; fixme)

`assertBeamQuant` - *l* (pair) *r* (pair)
: (undocumented; fixme)

`assertBeamSlope` - *comp* (procedure)
: (undocumented; fixme)

`autochange` - *music* (music)
: (undocumented; fixme)

`balloonGrobText` - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)
: (undocumented; fixme)

`balloonText` - *offset* (pair of numbers) *text* (markup)
: (undocumented; fixme)

`bar` - *type* (string)
: (undocumented; fixme)

`barNumberCheck` - *n* (integer)
: (undocumented; fixme)

`bendAfter` - *delta* (unknown)
: (undocumented; fixme)

`breathe` (undocumented; fixme)

`clef` - *type* (string)
: Set the current clef.

`cueDuring` - *what* (string) *dir* (direction) *main-music* (music)
: (undocumented; fixme)

`displayLilyMusic` - *music* (music)
: (undocumented; fixme)

`displayMusic` - *music* (music)
(undocumented; fixme)

`endSpanners` - *music* (music)
(undocumented; fixme)

`featherDurations` - *factor* (moment) *argument* (music)
Rearrange durations in ARGUMENT so there is an acceleration/deceleration.

`grace` - *music* (music)
(undocumented; fixme)

`includePageLayoutFile`
If page breaks and tweak dump is not asked, and the file <basename>-page-layout.ly exists, include it.

`instrumentSwitch` - *name* (string)
(undocumented; fixme)

`keepWithTag` - *tag* (symbol) *music* (music)
(undocumented; fixme)

`killCues` - *music* (music)
(undocumented; fixme)

`label` - *label* (symbol)
Place a bookmarking label, either at top-level or inside music.

`makeClusters` - *arg* (music)
(undocumented; fixme)

`musicMap` - *proc* (procedure) *mus* (music)
(undocumented; fixme)

`noPageBreak`
Forbid a page break. May be used at toplevel (ie between scores or markups), or inside a score.

`noPageTurn`
Forbid a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

`octaveCheck` - *pitch-note* (music)
octave check

`oldaddlyrics` - *music* (music) *lyrics* (music)
(undocumented; fixme)

`ottava` - *octave* (number)
set the octavation

`overrideProperty` - *name* (string) *property* (symbol) *value* (any type)
Set *property* to *value* in all grobs named *name*. The *name* argument is a string of the form `"Context.GrobName"` or `"GrobName"`

`pageBreak`
Force a page break. May be used at toplevel (ie between scores or markups), or inside a score.

`pageTurn` Force a page turn between two scores or top-level markups.

`parallelMusic` - *voice-ids* (list) *music* (music)
Define parallel music sequences, separated by '`|`' (bar check signs), and assign them to the identifiers provided in *voice-ids*.

voice-ids: a list of music identifiers (symbols containing only letters)

music: a music sequence, containing BarChecks as limiting expressions.

Example:

```
\parallelMusic #'(A B C) {
  c c | d d | e e |
  d d | e e | f f |
}
<==>
  A = { c c | d d | }
  B = { d d | e e | }
  C = { e e | f f | }
```

`parenthesize` - *arg* (music)
Tag *arg* to be parenthesized.

`partcombine` - *part1* (music) *part2* (music)
(undocumented; fixme)

`pitchedTrill` - *main-note* (music) *secondary-note* (music)
(undocumented; fixme)

`pointAndClickOff`
(undocumented; fixme)

`pointAndClickOn`
(undocumented; fixme)

`quoteDuring` - *what* (string) *main-music* (music)
(undocumented; fixme)

`removeWithTag` - *tag* (symbol) *music* (music)
(undocumented; fixme)

`resetRelativeOctave` - *reference-note* (music)
Set the octave inside a \relative section.

`rightHandFinger` - *finger* (number or string)
Define a StrokeFingerEvent

`scaleDurations` - *fraction* (pair of numbers) *music* (music)
(undocumented; fixme)

`scoreTweak` - *name* (string)
Include the score tweak, if exists.

`shiftDurations` - *dur* (integer) *dots* (integer) *arg* (music)
`spacingTweaks` - *parameters* (list)
Set the system stretch, by reading the 'system-stretch property of the 'parameters' assoc list.

`storePredefinedDiagram` - *chord* (music) *tuning* (pair) *diagram-definition* (unknown)
Add predefined fret diagram defined by `diagram-definition` for the chord pitches `chord` and the stringTuning `tuning`.

`tag` - *tag* (symbol) *arg* (music)
Add *tag* to the `tags` property of *arg*.

`tocItem` - *text* (markup)
: Add a line to the table of content, using the `tocItemMarkup` paper variable markup

`transposedCueDuring` - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)
: Insert notes from the part *what* into a voice called `cue`, using the transposition defined by *pitch-note*. This happens simultaneously with *main-music*, which is usually a rest. The argument *dir* determines whether the cue notes should be notated as a first or second voice.

`transposition` - *pitch-note* (music)
: Set instrument transposition

`tweak` - *sym* (symbol) *val* (any type) *arg* (music)
: Add `sym . val` to the `tweaks` property of *arg*.

`unfoldRepeats` - *music* (music)
: (undocumented; fixme)

`withMusicProperty` - *sym* (symbol) *val* (any type) *music* (music)
: Set *sym* to *val* in *music*.

B.15 Funciones de Scheme

`dispatcher` *x* [Función]
Is *x* a `Dispatcher` object?

`listener` *x* [Función]
Is *x* a `Listener` object?

`ly:add-file-name-alist` *alist* [Función]
Add mappings for error messages from *alist*.

`ly:add-interface` *a b c* [Función]
Add an interface description.

`ly:add-listener` *list disp cl* [Función]
Add the listener *list* to the dispatcher *disp*. Whenever *disp* hears an event of class *cl*, it is forwarded to *list*.

`ly:add-option` *sym val description* [Función]
Add a program option *sym* with default *val*.

`ly:all-grob-interfaces` [Función]
Get a hash table with all interface descriptions.

`ly:all-options` [Función]
Get all option settings in an alist.

`ly:all-stencil-expressions` [Función]
Return all symbols recognized as stencil expressions.

`ly:assoc-get` *key alist default-value* [Función]
Return value if *key* in *alist*, else `default-value` (or `#f` if not specified).

`ly:book-add-bookpart!` *book-smob book-part* [Función]
Add *book_part* to *book-smob* book part list.

`ly:book-add-score!` *book-smob score* [Función]
Add *score* to *book-smob* score list.

`ly:book-process` *book-smob default-paper default-layout output* [Función]
Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).

`ly:book-process-to-systems` *book-smob default-paper default-layout output* [Función]
Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).

`ly:box?` *x* [Función]
Is *x* a `Box` object?

`ly:bp` *num* [Función]
num bigpoints (1/72th inch).

`ly:bracket` *a iv t p* [Función]
Make a bracket in direction *a*. The extent of the bracket is given by *iv*. The wings protrude by an amount of *p*, which may be negative. The thickness is given by *t*.

`ly:broadcast` *disp ev* [Función]
Send the stream event *ev* to the dispatcher *disp*.

`ly:camel-case->lisp-identifier` *name-sym* [Función]
Convert `FooBar_Bla` to `foo-bar-bla` style symbol.

`ly:chain-assoc-get` *key achain dfault* [Función]
Return value for *key* from a list of alists *achain*. If no entry is found, return *dfault* or `#f` if no *dfault* is specified.

`ly:clear-anonymous-modules` [Función]
Plug a GUILE 1.6 and 1.7 memory leak by breaking a weak reference pointer cycle explicitly.

`ly:cm` *num* [Función]
num cm.

`ly:command-line-code` [Función]
The Scheme code specified on command-line with '`-e`'.

`ly:command-line-options` [Función]
The Scheme options specified on command-line with '`-d`'.

`ly:command-line-verbose?` [Función]
Was `be_verbose_global` set?

`ly:connect-dispatchers` *to from* [Función]
Make the dispatcher *to* listen to events from *from*.

`ly:context-event-source` *context* [Función]
Return `event-source` of context *context*.

`ly:context-events-below` *context* [Función]
Return a `stream-distributor` that distributes all events from *context* and all its subcontexts.

`ly:context-find` *context name* [Función]
Find a parent of *context* that has name or alias *name*. Return `#f` if not found.

`ly:context-grob-definition` *context name* [Función]
Return the definition of *name* (a symbol) within *context* as an alist.

`ly:context-id` *context* [Función]
Return the ID string of *context*, i.e., for `\context Voice = one ...` return the string `one`.

`ly:context-name` *context* [Función]
Return the name of *context*, i.e., for `\context Voice = one ...` return the symbol `Voice`.

`ly:context-now` *context* [Función]
Return `now-moment` of context *context*.

`ly:context-parent` *context* [Función]
Return the parent of *context*, `#f` if none.

`ly:context-property` *c name* [Función]
Return the value of *name* from context *c*.

`ly:context-property-where-defined` *context name* [Función]
Return the context above *context* where *name* is defined.

`ly:context-pushpop-property` *context grob eltprop val* [Función]
Do a single `\override` or `\revert` operation in *context*. The grob definition *grob* is extended with *eltprop* (if *val* is specified) or reverted (if unspecified).

`ly:context-set-property!` *context name val* [Función]
Set value of property *name* in context *context* to *val*.

`ly:context-unset-property` *context name* [Función]
Unset value of property *name* in context *context*.

`ly:context?` *x* [Función]
Is *x* a `Context` object?

`ly:default-scale` [Función]
Get the global default scale.

`ly:dimension?` *d* [Función]
Return *d* as a number. Used to distinguish length variables from normal numbers.

`ly:dir?` *s* [Función]
A type predicate. The direction `s` is `-1`, `0` or `1`, where `-1` represents left or down and `1` represents right or up.

`ly:duration->string` *dur* [Función]
Convert *dur* to a string.

`ly:duration-dot-count` *dur* [Función]
Extract the dot count from *dur*.

`ly:duration-factor` *dur* [Función]
Extract the compression factor from *dur*. Return it as a pair.

`ly:duration-length` *dur* [Función]
The length of the duration as a `moment`.

`ly:duration-log` *dur* [Función]
Extract the duration log from *dur*.

`ly:duration<?` *p1 p2* [Función]
Is *p1* shorter than *p2*?

`ly:duration?` *x* [Función]
Is *x* a `Duration` object?

`ly:effective-prefix` [Función]
Return effective prefix.

`ly:error` *str rest* [Función]
A Scheme callable function to issue the error *str*. The error is formatted with `format` and *rest*.

`ly:eval-simple-closure` *delayed closure scm-start scm-end* [Función]
Evaluate a simple *closure* with the given *delayed* argument. If *scm-start* and *scm-end* are defined, evaluate it purely with those start and end points.

`ly:event-deep-copy` *m* [Función]
Copy *m* and all sub expressions of *m*.

`ly:event-property` *sev sym* [Función]
Get the property *sym* of stream event *mus*. If *sym* is undefined, return `'()`.

`ly:event-set-property!` *ev sym val* [Función]
Set property *sym* in event *ev* to *val*.

`ly:expand-environment` *str* [Función]
Expand `$VAR` and `${VAR}` in *str*.

`ly:export` *arg* [Función]
Export a Scheme object to the parser so it is treated as an identifier.

`ly:find-accidentals-simple` *keysig pitch-scm barnum laziness octaveness* [Función]
Checks the need for an accidental and a 'restore' accidental against a key signature. The *laziness* is the number of bars for which reminder accidentals are used (ie. if *laziness* is zero, we only cancel accidentals in the same bar; if *laziness* is three, we cancel accidentals up to three bars after they first appear. *octaveness* is either `'same-octave` or `'any-octave` and it specifies whether accidentals should be canceled in different octaves.

`ly:find-file` *name* [Función]
Return the absolute file name of *name*, or `#f` if not found.

`ly:font-config-add-directory` *dir* [Función]
Add directory *dir* to FontConfig.

`ly:font-config-add-font` *font* [Función]
Add font *font* to FontConfig.

`ly:font-config-display-fonts` [Función]
Dump a list of all fonts visible to FontConfig.

`ly:font-config-get-font-file` *name* [Función]
Get the file for font *name*.

`ly:font-design-size` *font* [Función]
Given the font metric *font*, return the design size, relative to the current output-scale.

`ly:font-file-name` *font* [Función]
Given the font metric *font*, return the corresponding file name.

`ly:font-get-glyph` *font name* [Función]
Return a stencil from *font* for the glyph named *name*. If the glyph is not available, return an empty stencil.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Aybabtu fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-glyph-name-to-charcode` *font name* [Función]
Return the character code for glyph *name* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Aybabtu fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-glyph-name-to-index` *font name* [Función]
Return the index for *name* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Aybabtu fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-index-to-charcode` *font index* [Función]
Return the character code for *index* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Aybabtu fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-magnification` *font* [Función]
Given the font metric *font*, return the magnification, relative to the current output-scale.

`ly:font-metric?` *x* [Función]
Is *x* a `Font_metric` object?

`ly:font-name` *font* [Función]
Given the font metric *font*, return the corresponding name.

`ly:font-sub-fonts` *font* [Función]
Given the font metric *font* of an OpenType font, return the names of the subfonts within *font*.

`ly:format` *str rest* [Función]
LilyPond specific format, supporting `~a` and `~[0-9]f`.

`ly:format-output` *context* [Función]
Given a global context in its final state, process it and return the `Music_output` object in its final state.

`ly:get-all-function-documentation` [Función]
Get a hash table with all LilyPond Scheme extension functions.

`ly:get-all-translators` [Función]
Return a list of all translator objects that may be instantiated.

`ly:get-glyph` *font index* [Función]
Retrieve a stencil for the glyph numbered *index* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Aybabtu fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:get-listened-event-classes` [Función]
Return a list of all event classes that some translator listens to.

`ly:get-option` *var* [Función]
Get a global option setting.

`ly:gettext` *original* [Función]
A Scheme wrapper function for `gettext`.

`ly:grob-alist-chain` *grob global* [Función]
Get an alist chain for grob *grob*, with *global* as the global default. If unspecified, `font-defaults` from the layout block is taken.

`ly:grob-array-length` *grob-arr* [Función]
Return the length of *grob-arr*.

`ly:grob-array-ref` *grob-arr index* [Función]
Retrieve the *index*th element of *grob-arr*.

`ly:grob-array?` *x* [Función]
Is *x* a `Grob_array` object?

`ly:grob-basic-properties` *grob* [Función]
Get the immutable properties of *grob*.

`ly:grob-common-refpoint` *grob other axis* [Función]
Find the common refpoint of *grob* and *other* for *axis*.

`ly:grob-common-refpoint-of-array` *grob others axis* [Función]
Find the common refpoint of *grob* and *others* (a grob-array) for *axis*.

`ly:grob-default-font` *grob* [Función]
Return the default font for grob *gr*.

`ly:grob-extent` *grob refp axis* [Función]
Get the extent in *axis* direction of *grob* relative to the grob *refp*.

`ly:grob-interfaces` *grob* [Función]
Return the interfaces list of grob *grob*.

`ly:grob-layout` *grob* [Función]
Get `\layout` definition from grob *grob*.

`ly:grob-object` *grob sym* [Función]
Return the value of a pointer in grob *g* of property *sym*. It returns `'()` (end-of-list) if *sym* is undefined in *g*.

`ly:grob-original` *grob* [Función]
Return the unbroken original grob of *grob*.

`ly:grob-parent` *grob axis* [Función]
Get the parent of *grob*. *axis* is 0 for the X-axis, 1 for the Y-axis.

`ly:grob-pq<?` *a b* [Función]
Compare two grob priority queue entries. This is an internal function.

`ly:grob-properties` *grob* [Función]
Get the mutable properties of *grob*.

`ly:grob-property` *grob sym defflt* [Función]
Return the value of a value in grob *g* of property *sym*. It returns `'()` (end-of-list) or *defflt* (if specified) if *sym* is undefined in *g*.

`ly:grob-property-data` *grob sym* [Función]
Retrieve *sym* for *grob* but don't process callbacks.

`ly:grob-relative-coordinate` *grob refp axis* [Función]
Get the coordinate in *axis* direction of *grob* relative to the grob *refp*.

`ly:grob-robust-relative-extent` *grob refp axis* [Función]
Get the extent in *axis* direction of *grob* relative to the grob *refp*, or `(0,0)` if empty.

`ly:grob-script-priority-less` *a b* [Función]
Compare two grobs by script priority. For internal use.

`ly:grob-set-property!` *grob sym val* [Función]
Set *sym* in grob *grob* to value *val*.

`ly:grob-staff-position` *sg* [Función]
Return the Y-position of *sg* relative to the staff.

`ly:grob-suicide!` *grob* [Función]
Kill *grob*.

`ly:grob-system` *grob* [Función]
Return the system grob of *grob*.

`ly:grob-translate-axis!` *grob d a* [Función]
Translate *g* on axis *a* over distance *d*.

`ly:grob?` *x* [Función]
Is *x* a `Grob` object?

`ly:gulp-file` *name size* [Función]
Read the file *name*, and return its contents in a string. The file is looked up using the search path.

`ly:hash-table-keys` *tab* [Función]
Return a list of keys in *tab*.

`ly:inch` *num* [Función]
num inches.

`ly:input-both-locations` *sip* [Función]
Return input location in *sip* as `(file-name first-line first-column last-line last-column)`.

`ly:input-file-line-char-column` *sip* [Función]
Return input location in *sip* as `(file-name line char column)`.

`ly:input-location?` *x* [Función]
Is *x* an `input-location`?

`ly:input-message` *sip msg rest* [Función]
Print *msg* as a GNU compliant error message, pointing to the location in *sip*. *msg* is interpreted similar to `format`'s argument, using *rest*.

`ly:interpret-music-expression` *mus ctx* [Función]
Interpret the music expression *mus* in the global context *ctx*. The context is returned in its final state.

`ly:interpret-stencil-expression` *expr func arg1 offset* [Función]
Parse *expr*, feed bits to *func* with first arg *arg1* having offset *offset*.

`ly:intlog2` *d* [Función]
The 2-logarithm of $1/d$.

`ly:is-listened-event-class` *sym* [Función]
Is *sym* a listened event class?

`ly:item-break-dir` *it* [Función]
The break status direction of item *it*. `-1` means end of line, `0` unbroken, and `1` beginning of line.

`ly:item?` *g* [Función]
Is *g* an `Item` object?

`ly:iterator?` *x* [Función]
Is *x* a `Music_iterator` object?

`ly:lexer-keywords` *lexer* [Función]
Return a list of `(KEY . CODE)` pairs, signifying the LilyPond reserved words list.

`ly:lily-lexer?` *x* [Función]
Is *x* a `Lily_lexer` object?

`ly:lily-parser?` *x* [Función]
Is *x* a `Lily_parser` object?

`ly:make-book` *paper header scores* [Función]
Make a `\book` of *paper* and *header* (which may be `#f` as well) containing `\scores`.

`ly:make-book-part` *scores* [Función]
Make a `\bookpart` containing `\scores`.

`ly:make-dispatcher` [Función]
Return a newly created dispatcher.

`ly:make-duration` *length dotcount num den* [Función]
length is the negative logarithm (base 2) of the duration: 1 is a half note, 2 is a quarter note, 3 is an eighth note, etc. The number of dots after the note is given by the optional argument *dotcount*.

The duration factor is optionally given by *num* and *den*.

A duration is a musical duration, i.e., a length of time described by a power of two (whole, half, quarter, etc.) and a number of augmentation dots.

`ly:make-global-context` *output-def* [Función]
Set up a global interpretation context, using the output block *output_def*. The context is returned.

`ly:make-global-translator` *global* [Función]
Create a translator group and connect it to the global context *global*. The translator group is returned.

`ly:make-listener` *callback* [Función]
Create a listener. Any time the listener hears an object, it will call *callback* with that object.

callback should take exactly one argument.

`ly:make-moment` *n d gn gd* [Función]
Create the rational number with main timing n/d, and optional grace timing gn/gd.

A *moment* is a point in musical time. It consists of a pair of rationals (m, g), where m is the timing for the main notes, and g the timing for grace notes. In absence of grace notes, g is zero.

`ly:make-music` *props* [Función]
Make a C++ `Music` object and initialize it with *props*.

This function is for internal use and is only called by `make-music`, which is the preferred interface for creating music objects.

`ly:make-music-function` *signature func* [Función]
Make a function to process music, to be used for the parser. `func` is the function, and `signature` describes its arguments. `signature` is a list containing either `ly:music?` predicates or other type predicates.

`ly:make-output-def` [Función]
Make an output definition.

`ly:make-page-label-marker` *label* [Función]
Return page marker with label.

`ly:make-page-permission-marker` *symbol permission* [Función]
Return page marker with page breaking and turning permissions.

`ly:make-pango-description-string` *chain size* [Función]
Make a `PangoFontDescription` string for the property alist *chain* at size *size*.

`ly:make-paper-outputter` *port format* [Función]
Create an outputter that evaluates within `output-`*format*, writing to *port*.

`ly:make-pitch` *octave note alter* [Función]
octave is specified by an integer, zero for the octave containing middle C. *note* is a number from 0 to 6, with 0 corresponding to pitch C and 6 corresponding to pitch B. *alter* is a rational number of whole tones for alteration.

`ly:make-prob` *type init rest* [Función]
Create a `Prob` object.

`ly:make-scale` *steps* [Función]
Create a scale. The argument is a vector of rational numbers, each of which represents the number of tones of a pitch above the tonic.

`ly:make-score` *music* [Función]
Return score with *music* encapsulated in *score*.

`ly:make-simple-closure` *expr* [Función]
Make a simple closure. *expr* should be form of `(func a1 A2 ...)`, and will be invoked as `(func delayed-arg a1 a2 ...)`.

`ly:make-stencil` *expr xext yext* [Función]
Stencils are device independent output expressions. They carry two pieces of information:

1. A specification of how to print this object. This specification is processed by the output backends, for example '`scm/output-ps.scm`'.
2. The vertical and horizontal extents of the object, given as pairs. If an extent is unspecified (or if you use `(1000 . -1000)` as its value), it is taken to be empty.

`ly:make-stream-event` *cl proplist* [Función]
Create a stream event of class *cl* with the given mutable property list.

`ly:message` *str rest* [Función]
A Scheme callable function to issue the message *str*. The message is formatted with `format` and *rest*.

`ly:minimal-breaking` *pb* [Función]
Break (pages and lines) the `Paper_book` object *pb* without looking for optimal spacing: stack as many lines on a page before moving to the next one.

`ly:mm` *num* [Función]
num mm.

`ly:module->alist` *mod* [Función]
Dump the contents of module *mod* as an alist.

`ly:module-copy` *dest src* [Función]
Copy all bindings from module *src* into *dest*.

`ly:modules-lookup` *modules sym def* [Función]
Look up *sym* in the list *modules*, returning the first occurence. If not found, return *def* or `#f` if *def* isn't specified.

`ly:moment-add` *a b* [Función]
Add two moments.

`ly:moment-div` *a b* [Función]
Divide two moments.

`ly:moment-grace-denominator` *mom* [Función]
Extract denominator from grace timing.

`ly:moment-grace-numerator` *mom* [Función]
Extract numerator from grace timing.

`ly:moment-main-denominator` *mom* [Función]
Extract denominator from main timing.

`ly:moment-main-numerator` *mom* [Función]
Extract numerator from main timing.

`ly:moment-mod` *a b* [Función]
Modulo of two moments.

`ly:moment-mul` *a b* [Función]
Multiply two moments.

`ly:moment-sub` *a b* [Función]
Subtract two moments.

`ly:moment<?` *a b* [Función]
Compare two moments.

`ly:moment?` *x* [Función]
Is *x* a `Moment` object?

`ly:music-compress` *m factor* [Función]
Compress music object *m* by moment *factor*.

`ly:music-deep-copy` *m* [Función]
Copy *m* and all sub expressions of *m*.

`ly:music-duration-compress` *mus fact* [Función]
Compress *mus* by factor *fact*, which is a `Moment`.

`ly:music-duration-length` *mus* [Función]
Extract the duration field from *mus* and return the length.

`ly:music-function-extract` *x* [Función]
Return the Scheme function inside *x*.

`ly:music-function?` *x* [Función]
Is *x* a `music-function`?

`ly:music-length` *mus* [Función]
Get the length of music expression *mus* and return it as a `Moment` object.

`ly:music-list?` *lst* [Función]
Type predicate: Return true if *lst* is a list of music objects.

`ly:music-mutable-properties` *mus* [Función]
Return an alist containing the mutable properties of *mus*. The immutable properties are not available, since they are constant and initialized by the `make-music` function.

`ly:music-output?` *x* [Función]
Is *x* a `Music_output` object?

`ly:music-property` *mus sym dfault* [Función]
Get the property *sym* of music expression *mus*. If *sym* is undefined, return `'()`.

`ly:music-set-property!` *mus sym val* [Función]
Set property *sym* in music expression *mus* to *val*.

`ly:music-transpose` *m p* [Función]
Transpose *m* such that central C is mapped to *p*. Return *m*.

`ly:music?` *obj* [Función]
Type predicate.

`ly:note-head::stem-attachment` *font-metric glyph-name* [Función]
Get attachment in *font-metric* for attaching a stem to notehead *glyph-name*.

`ly:number->string` *s* [Función]
Convert *num* to a string without generating many decimals.

`ly:optimal-breaking` *pb* [Función]
Optimally break (pages and lines) the `Paper_book` object *pb* to minimize badness in bother vertical and horizontal spacing.

`ly:option-usage` [Función]
Print `ly:set-option` usage.

`ly:otf->cff` *otf-file-name* [Función]
Convert the contents of an OTF file to a CFF file, returning it as a string.

`ly:otf-font-glyph-info` *font glyph* [Función]
Given the font metric *font* of an OpenType font, return the information about named glyph *glyph* (a string).

`ly:otf-font-table-data` *font tag* [Función]
Extract a table *tag* from *font*. Return empty string for non-existent *tag*.

`ly:otf-font?` *font* [Función]
Is *font* an OpenType font?

`ly:otf-glyph-list` *font* [Función]
Return a list of glyph names for *font*.

`ly:output-def-clone` *def* [Función]
Clone output definition *def*.

`ly:output-def-lookup` *pap sym def* [Función]
Look up *sym* in the *pap* output definition (e.g., `\paper`). Return the value or *def* (which defaults to `'()`) if undefined.

`ly:output-def-parent` *def* [Función]
Get the parent output definition of *def*.

`ly:output-def-scope` *def* [Función]
Get the variable scope inside *def*.

`ly:output-def-set-variable!` *def sym val* [Función]
Set an output definition *def* variable *sym* to *val*.

`ly:output-def?` *def* [Función]
Is *def* a layout definition?

`ly:output-description` *output-def* [Función]
Return the description of translators in *output-def*.

`ly:output-formats` [Función]
Formats passed to '`--format`' as a list of strings, used for the output.

`ly:outputter-close` *outputter* [Función]
Close port of *outputter*.

`ly:outputter-dump-stencil` *outputter stencil* [Función]
Dump stencil *expr* onto *outputter*.

`ly:outputter-dump-string` *outputter str* [Función]
Dump *str* onto *outputter*.

`ly:outputter-output-scheme` *outputter expr* [Función]
Eval *expr* in module of *outputter*.

`ly:outputter-port` *outputter* [Función]
Return output port for *outputter*.

`ly:page-marker?` *x* [Función]
Is *x* a `Page_marker` object?

`ly:page-turn-breaking` *pb* [Función]
Optimally break (pages and lines) the `Paper_book` object *pb* such that page turns only happen in specified places, returning its pages.

`ly:pango-font-physical-fonts` *f* [Función]
Return alist of `(ps-name file-name font-index)` lists for Pango font *f*.

`ly:pango-font?` *f* [Función]
Is *f* a pango font?

`ly:paper-book-pages` *pb* [Función]
Return pages in book *pb*.

`ly:paper-book-paper` *pb* [Función]
Return pages in book *pb*.

`ly:paper-book-performances` *paper-book* [Función]
Return performances in book *paper-book*.

`ly:paper-book-scopes` *book* [Función]
Return scopes in layout book *book*.

`ly:paper-book-systems` *pb* [Función]
Return systems in book *pb*.

`ly:paper-book?` *x* [Función]
Is *x* a `Paper_book` object?

`ly:paper-fonts` *bp* [Función]
Return fonts from the `\paper` block *bp*.

`ly:paper-get-font` *paper-smob chain* [Función]
Return a font metric satisfying the font-qualifiers in the alist chain *chain*. (An alist chain is a list of alists, containing grob properties.)

`ly:paper-get-number` *layout-smob name* [Función]
Return the layout variable *name*.

`ly:paper-outputscale` *bp* [Función]
Get output-scale for *bp*.

`ly:paper-score-paper-systems` *paper-score* [Función]
Return vector of `paper_system` objects from *paper-score*.

`ly:paper-system-minimum-distance` *sys1 sys2* [Función]
Measure the minimum distance between these two paper-systems, using their stored skylines if possible and falling back to their extents otherwise.

`ly:paper-system?` *obj* [Función]
Type predicate.

`ly:parse-file` *name* [Función]
Parse a single `.ly` file. Upon failure, throw `ly-file-failed` key.

`ly:parser-clear-error` *parser* [Función]
Clear the error flag for the parser.

`ly:parser-clone` *parser-smob* [Función]
Return a clone of *parser-smob*.

`ly:parser-define!` *parser-smob symbol val* [Función]
Bind *symbol* to *val* in *parser-smob*'s module.

`ly:parser-error` *parser msg input* [Función]
Display an error message and make the parser fail.

`ly:parser-has-error?` *parser* [Función]
Does *parser* have an error flag?

`ly:parser-lexer` *parser-smob* [Función]
Return the lexer for *parser-smob*.

`ly:parser-lookup` *parser-smob symbol* [Función]
Look up *symbol* in *parser-smob*'s module. Return `'()` if not defined.

`ly:parser-output-name` *parser* [Función]
Return the base name of the output file.

`ly:parser-parse-string` *parser-smob ly-code* [Función]
Parse the string *ly-code* with *parser-smob*. Upon failure, throw `ly-file-failed` key.

`ly:parser-set-note-names` *parser names* [Función]
Replace current note names in *parser*. *names* is an alist of symbols. This only has effect if the current mode is notes.

`ly:performance-write` *performance filename* [Función]
Write *performance* to *filename*.

`ly:pfb->pfa` *pfb-file-name* [Función]
Convert the contents of a PFB file to PFA.

`ly:pitch-alteration` *pp* [Función]
Extract the alteration from pitch *pp*.

`ly:pitch-diff` *pitch root* [Función]
Return pitch *delta* such that *pitch* transposed by *delta* equals *root*.

`ly:pitch-negate` *p* [Función]
Negate *p*.

`ly:pitch-notename` *pp* [Función]
Extract the note name from pitch *pp*.

`ly:pitch-octave` *pp* [Función]
Extract the octave from pitch *pp*.

`ly:pitch-quartertones` *pp* [Función]
Calculate the number of quarter tones of *pp* from middle C.

`ly:pitch-semitones` *pp* [Función]
Calculate the number of semitones of *pp* from middle C.

`ly:pitch-steps` *p* [Función]
Number of steps counted from middle C of the pitch *p*.

`ly:pitch-transpose` *p delta* [Función]
Transpose *p* by the amount *delta*, where *delta* is relative to middle C.

`ly:pitch<?` *p1 p2* [Función]
Is *p1* lexicographically smaller than *p2*?

`ly:pitch?` *x* [Función]
Is *x* a `Pitch` object?

`ly:position-on-line?` *sg spos* [Función]
Return whether *pos* is on a line of the staff associated with the the grob *sg* (even on an extender line).

`ly:prob-immutable-properties` *prob* [Función]
Retrieve an alist of mutable properties.

`ly:prob-mutable-properties` *prob* [Función]
Retrieve an alist of mutable properties.

`ly:prob-property` *obj sym dfault* [Función]
Return the value for *sym*.

`ly:prob-property?` *obj sym* [Función]
Is boolean prop *sym* set?

`ly:prob-set-property!` *obj sym value* [Función]
Set property *sym* of *obj* to *value*.

`ly:prob-type?` *obj type* [Función]
Is *obj* the specified prob-type?

`ly:prob?` *x* [Función]
Is *x* a `Prob` object?

`ly:programming-error` *str rest* [Función]
A Scheme callable function to issue the internal warning *str*. The message is formatted with `format` and *rest*.

`ly:progress` *str rest* [Función]
A Scheme callable function to print progress *str*. The message is formatted with `format` and *rest*.

`ly:property-lookup-stats` *sym* [Función]
Return hash table with a property access corresponding to *sym*. Choices are `prob`, `grob`, and `context`.

`ly:protects` [Función]
Return hash of protected objects.

`ly:pt` *num* [Función]
num printer points.

`ly:register-stencil-expression` *symbol* [Función]
Add *symbol* as head of a stencil expression.

`ly:relative-group-extent` *elements common axis* [Función]
Determine the extent of *elements* relative to *common* in the *axis* direction.

`ly:reset-all-fonts` [Función]
Forget all about previously loaded fonts.

`ly:round-filled-box` *xext yext blot* [Función]
Make a `Stencil` object that prints a black box of dimensions *xext*, *yext* and roundness *blot*.

`ly:round-filled-polygon` *points blot* [Función]
Make a `Stencil` object that prints a black polygon with corners at the points defined by *points* (list of coordinate pairs) and roundness *blot*.

`ly:run-translator` *mus output-def* [Función]
Process *mus* according to *output-def*. An interpretation context is set up, and *mus* is interpreted with it. The context is returned in its final state.

Optionally, this routine takes an object-key to to uniquely identify the score block containing it.

`ly:score-add-output-def!` *score def* [Función]
Add an output definition *def* to *score*.

`ly:score-embedded-format` *score layout* [Función]
Run *score* through *layout* (an output definition) scaled to correct output-scale already, returning a list of layout-lines. This function takes an optional `Object_key` argument.

`ly:score-error?` *score* [Función]
Was there an error in the score?

`ly:score-header` *score* [Función]
Return score header.

`ly:score-music` *score* [Función]
Return score music.

`ly:score-output-defs` *score* [Función]
All output definitions in a score.

`ly:score-set-header!` *score module* [Función]
Set the score header.

`ly:score?` *x* [Función]
Is *x* a `Score` object?

`ly:set-default-scale` *scale* [Función]
Set the global default scale.

`ly:set-grob-creation-callback` *cb* [Función]
Specify a procedure that will be called every time a new grob is created. The callback will receive as arguments the grob that was created, the name of the C++ source file that caused the grob to be created, and the corresponding line number in the C++ source file.

`ly:set-grob-modification-callback` *cb* [Función]
Specify a procedure that will be called every time LilyPond modifies a grob property. The callback will receive as arguments the grob that is being modified, the name of the C++ file in which the modification was requested, the line number in the C++ file in which the modification was requested, the name of the function in which the modification was requested, the property to be changed, and the new value for the property.

`ly:set-middle-C!` *context* [Función]
Set the `middleCPosition` variable in *context* based on the variables `middleCClefPosition` and middleCOffset.

`ly:set-option` *var val* [Función]
Set a program option.

`ly:set-point-and-click` *what* [Función]
Deprecated.

`ly:set-property-cache-callback` *cb* [Función]
Specify a procedure that will be called whenever lilypond calculates a callback function and caches the result. The callback will receive as arguments the grob whose property it is, the name of the property, the name of the callback that calculated the property, and the new (cached) value of the property.

`ly:simple-closure?` *clos* [Función]
Type predicate.

`ly:skyline-pair?` *x* [Función]
Is *x* a `Skyline_pair` object?

`ly:skyline?` *x* [Función]
Is *x* a `Skyline` object?

`ly:smob-protects` [Función]
Return LilyPond's internal smob protection list.

`ly:solve-spring-rod-problem` *springs rods length ragged* [Función]
Solve a spring and rod problem for *count* objects, that are connected by *count*-1 *springs*, and an arbitrary number of *rods*. *count* is implicitly given by *springs* and *rods*. The *springs* argument has the format `(ideal, inverse_hook)` and *rods* is of the form `(idx1, idx2, distance)`.

length is a number, *ragged* a boolean.

The function returns a list containing the force (positive for stretching, negative for compressing and `#f` for non-satisfied constraints) followed by *spring-count*+1 positions of the objects.

`ly:source-file?` *x* [Función]
Is *x* a `Source_file` object?

`ly:spanner-bound` *slur dir* [Función]
Get one of the bounds of *slur*. *dir* is `-1` for left, and `1` for right.

`ly:spanner-broken-into` *spanner* [Función]
Return broken-into list for *spanner*.

`ly:spanner?` *g* [Función]
Is *g* a spanner object?

`ly:staff-symbol-line-thickness` *grob* [Función]
Returns the line-thickness of the staff associated with *grob*.

`ly:start-environment` [Función]
Return the environment (a list of strings) that was in effect at program start.

`ly:stderr-redirect` *file-name mode* [Función]
Redirect stderr to *file-name*, opened with *mode*.

`ly:stencil-add` *args* [Función]
Combine stencils. Takes any number of arguments.

`ly:stencil-aligned-to` *stil axis dir* [Función]
Align *stil* using its own extents. *dir* is a number. `-1` and `1` are left and right, respectively. Other values are interpolated (so `0` means the center).

`ly:stencil-combine-at-edge` *first axis direction second padding minimum* [Función]
Construct a stencil by putting *second* next to *first*. *axis* can be 0 (x-axis) or 1 (y-axis). *direction* can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with *padding* as extra space. If this puts the reference points closer than *minimum*, they are moved by the latter amount. *first* and *second* may also be `'()` or `#f`.

`ly:stencil-empty?` *stil* [Función]
Return whether *stil* is empty.

`ly:stencil-expr` *stil* [Función]
Return the expression of *stil*.

`ly:stencil-extent` *stil axis* [Función]
Return a pair of numbers signifying the extent of *stil* in *axis* direction (`0` or `1` for x and y axis, respectively).

`ly:stencil-fonts` *s* [Función]
Analyze *s*, and return a list of fonts used in *s*.

`ly:stencil-in-color` *stc r g b* [Función]
Put *stc* in a different color.

`ly:stencil-rotate` *stil angle x y* [Función]
Return a stencil *stil* rotated *angle* degrees around the relative offset (*x*, *y*). E.g. an offset of (-1, 1) will rotate the stencil around the left upper corner.

`ly:stencil-rotate-absolute` *stil angle x y* [Función]
Return a stencil *stil* rotated *angle* degrees around point (*x*, *y*), given in absolute coordinates.

`ly:stencil-translate` *stil offset* [Función]
Return a *stil*, but translated by *offset* (a pair of numbers).

`ly:stencil-translate-axis` *stil amount axis* [Función]
Return a copy of *stil* but translated by *amount* in *axis* direction.

`ly:stencil?` *x* [Función]
Is *x* a `Stencil` object?

`ly:stream-event?` *x* [Función]
Is *x* a `Stream_event` object?

`ly:string-substitute` *a b s* [Función]
Replace string *a* by string *b* in string *s*.

`ly:system-font-load` *name* [Función]
Load the OpenType system font '*`name`*`.otf`'. Fonts loaded with this command must contain three additional SFNT font tables called `LILC`, `LILF`, and `LILY`, needed for typesetting musical elements. Currently, only the Emmentaler and the Aybabtu fonts fulfill these requirements.

Note that only `ly:font-get-glyph` and derived code (like `\lookup`) can access glyphs from the system fonts; text strings are handled exclusively via the Pango interface.

`ly:system-print` *system* [Función]
Draw the system and return the prob containing its stencil.

`ly:system-stretch` *system amount-scm* [Función]
Stretch the system vertically by the given amount. This must be called before the system is drawn (for example with `ly:system-print`).

`ly:text-dimension` *font text* [Función]
Given the font metric in *font* and the string *text*, compute the extents of that text in that font. The return value is a pair of number-pairs.

`ly:text-interface::interpret-markup` [Función]
Convert a text markup into a stencil. Takes three arguments, *layout*, *props*, and *markup*.

layout is a `\layout` block; it may be obtained from a grob with `ly:grob-layout`. *props* is an alist chain, i.e. a list of alists. This is typically obtained with `(ly:grob-alist-chain (ly:layout-lookup layout 'text-font-defaults))`. *markup* is the markup text to be processed.

`ly:translator-description` *me* [Función]
Return an alist of properties of translator *me*.

`ly:translator-group?` *x* [Función]
Is *x* a `Translator_group` object?

`ly:translator-name` *trans* [Función]
Return the type name of the translator object *trans*. The name is a symbol.

`ly:translator?` *x* [Función]
Is *x* a `Translator` object?

`ly:transpose-key-alist` *l pit* [Función]
Make a new key alist of *l* transposed by pitch *pit*.

`ly:truncate-list!` *lst i* [Función]
Take at most the first *i* of list *lst*.

`ly:ttf->pfa` *ttf-file-name idx* [Función]
Convert the contents of a TrueType font file to PostScript Type 42 font, returning it as a string. The optional *idx* argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of *idx* is 0.

`ly:ttf-ps-name` *ttf-file-name idx* [Función]
Extract the PostScript name from a TrueType font. The optional *idx* argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of *idx* is 0.

`ly:unit` [Función]
Return the unit used for lengths as a string.
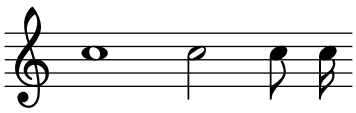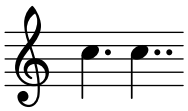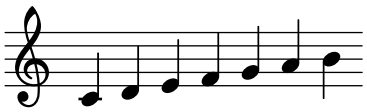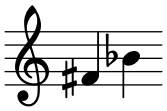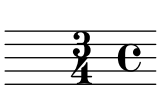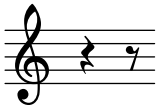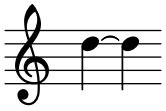
`ly:usage` [Función]
Print usage message.

`ly:version` [Función]
Return the current lilypond version as a list, e.g., `(1 3 127 uu1)`.

`ly:warning` *str rest* [Función]
A Scheme callable function to issue the warning `str`. The message is formatted with `format` and `rest`.

`ly:wide-char->utf-8` *wc* [Función]
Encode the Unicode codepoint *wc*, an integer, as UTF-8.
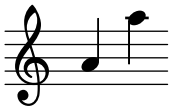
Apéndice C Hoja de referencia rápida

Sintaxis	Descripción	Ejemplo
`1 2 8 16`	duraciones	
`c4. c4..`	puntillos	
`c d e f g a b`	escala	
`fis bes`	alteración	
`\clef treble \clef bass`	claves	
`\time 3/4 \time 4/4`	compás	
`r4 r8`	silencio	
`d ~ d`	ligadura de unión	
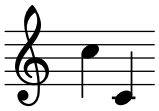
`\key es \major` armadura
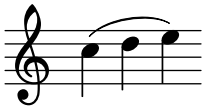


note`'` subir una octava
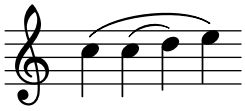


note`,` bajar una octava
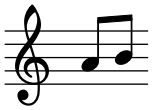


`c( d e)` ligadura de expresión



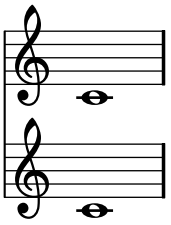
`c\( c( d) e\)` ligadura de fraseo
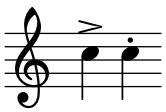


`a8[ b]` barra
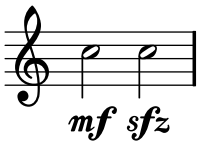


`<< \new Staff ... >>` más pentagramas
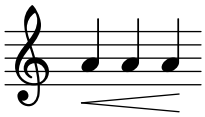


`c-> c-.` articulaciones
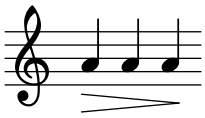


`c2\mf c\sfz` matices dinámicos
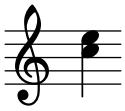


`a\< a a\!` crescendo
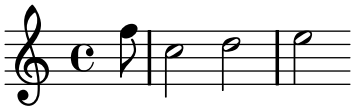


`a\> a a\!` decrescendo
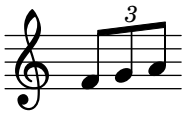


`< >` acorde
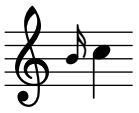


`\partial 8` anacrusa



`\times 2/3 {f g a}` tresillos



`\grace` mordentes

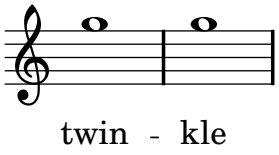


`\lyricmode { twinkle }` escribir la letra twinkle
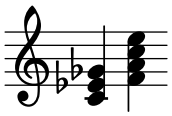
`\new Lyrics` imprimir la letra

twinkle

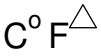
`twin -- kle` separador de sílabas
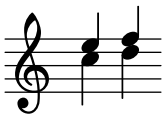


`\chordmode { c:dim f:maj7 }` acordes



`\context ChordNames` imprimir los nombres de los acordes



`<<{e f} \\ {c d}>>` polifonía



`s4 s8 s16` silencios de separación

Apéndice D Licencia de documentación libre de GNU

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other written document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of 'copyleft', which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The 'Document', below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as 'you'.

   A 'Modified Version' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A 'Secondary Section' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The 'Invariant Sections' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

   The 'Cover Texts' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

   A 'Transparent' copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file

format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not 'Transparent' is called 'Opaque'.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The 'Title Page' means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, 'Title Page' means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled 'History', and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled 'History' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the 'History' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled 'Acknowledgments' or 'Dedications', preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled 'Endorsements'. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as 'Endorsements' or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to

the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled 'Endorsements', provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled 'History' in the various original documents, forming one section entitled 'History'; likewise combine any sections entitled 'Acknowledgments', and any sections entitled 'Dedications'. You must delete all sections entitled 'Endorsements.'

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an 'aggregate', and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License 'or any later version' applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
  Copyright (C)  year  your name.
  Permission is granted to copy, distribute and/or modify this document
  under the terms of the GNU Free Documentation License, Version 1.1
  or any later version published by the Free Software Foundation;
  with the Invariant Sections being list their titles, with the
  Front-Cover Texts being list, and with the Back-Cover Texts being list.
  A copy of the license is included in the section entitled ‘GNU
Free Documentation License’
.
```

If you have no Invariant Sections, write ‘with no Invariant Sections’ instead of saying which ones are invariant. If you have no Front-Cover Texts, write ‘no Front-Cover Texts’ instead of ‘Front-Cover Texts being *list*’; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Apéndice E Índice de instrucciones de LilyPond

Este índice relaciona todas las instrucciones y palabras clave de LilyPond, con enlaces a aquellas secciones del manual que describen o se ocupan de su uso. Cada uno de los enlaces consta de dos partes. La primera parte apunta a la situación exacta del manual en que aparece la instrucción o palabra clave; la segunda parte apunta al comienzo de la sección correspondiente del manual en la que aparece la instrucción o palabra clave.

!

! 5

'

' 1

(

(begin * * * *) 58
(end * * * *) 58

,

, ... 1

-

- 84

.

. 32

/

/ 268
/+ 268

:

: 110

<

< 111
<...> 111

=

= 8

>

> 111

?

? 5

[

[..... 67

]

] 67

^

^ 268

_

_ 192, 196

\

\! 86
\(..... 93
\) 93
\< 86
\> 86
\abs-fontsize 476
\accepts 395, 396
\addChordShape 241
\addlyrics 194
\aeolian 15
\afterGrace 79
\aikenHeads 28
\allowPageTurn 356
\alternative 102
\AncientRemoveEmptyStaffContext 140
\applyContext 447
\applyOutput 447
\arpeggio 97, 98
\arpeggioArrowDown 97, 98
\arpeggioArrowUp 97, 98
\arpeggioBracket 98
\arpeggioNormal 97, 98
\arpeggioParenthesis 98
\arrow-head 182, 498
\ascendens 298
\ascendens 304
\auctum 298
\auctum 304
\augmentum 304
\autoBeamOff 67
\autoBeamOn 67
\autochange 210
\backslashed-digit 508
\balloonGrobText 161
\balloonLengthOff 161
\balloonLengthOn 161
\balloonText 161
\bar 70
\beam 498
\bendAfter 95
\bold 175, 476

\book ... 317, 318
\bookpart ... 318, 354
\box ... 180, 476
\bracket ... 90, 180, 498
\break ... 353
\breathe ... 94
\breve ... 31, 39
\cadenzaOff ... 49
\cadenzaOn ... 49
\caesura ... 296
\caps ... 477
\cavum ... 298
\cavum ... 304
\center-align ... 177, 485
\center-column ... 179, 485
\change ... 209
\char ... 508
\chordmode ... 4, 11, 238
\chords ... 270
\circle ... 180, 498
\clef ... 12
\cm ... 408
\column ... 179, 485
\column-lines ... 512
\combine ... 182, 486
\compressFullBarRests ... 42
\concat ... 486
\context ... 390
\cr ... 86
\crescHairpin ... 87
\crescTextCresc ... 87
\decr ... 86
\defaultTimeSignature ... 46
\deminutum ... 298
\deminutum ... 304
\denies ... 395, 396
\descendens ... 298
\descendens ... 304
\dimHairpin ... 87
\dimTextDecr ... 87
\dimTextDecresc ... 87
\dimTextDim ... 87
\dir-column ... 486
\displayLilyMusic ... 333, 439
\displayMusic ... 438
\divisioMaior ... 296
\divisioMaxima ... 296
\divisioMinima ... 296
\dorian ... 15
\dotsDown ... 32
\dotsNeutral ... 32
\dotsUp ... 32
\doubleflat ... 502
\doublesharp ... 502
\downbow ... 220
\draw-circle ... 182, 499
\draw-line ... 182, 499
\dynamic ... 90, 477
\dynamicDown ... 88
\dynamicNeutral ... 88
\dynamicUp ... 88
\easyHeadsOff ... 28
\easyHeadsOn ... 28
\epsfile ... 182, 499
\espressivo ... 87
\expandFullBarRests ... 42
\f ... 86
\featherDurations ... 69
\ff ... 86
\fff ... 86
\ffff ... 86
\fill-line ... 179, 487
\filled-box ... 182, 499
\finalis ... 296
\finger ... 155, 477
\flat ... 502
\flexa ... 304
\fontCaps ... 477
\fontsize ... 175, 477
\fp ... 86
\fraction ... 508
\frenchChords ... 273
\fret-diagram ... 231, 505
\fret-diagram-terse ... 232, 506
\fret-diagram-verbose ... 234, 506
\fromproperty ... 508
\general-align ... 178, 487
\germanChords ... 273
\glissando ... 96
\grace ... 78
\halign ... 177, 488
\harmonic ... 221
\harp-pedal ... 507
\hbracket ... 180, 500
\hcenter-in ... 489
\header ... 318
\hideKeySignature ... 262
\hideNotes ... 158
\hideStaffSwitch ... 211
\hspace ... 489
\huge ... 154, 177, 478
\improvisationOff ... 30
\improvisationOn ... 30
\in ... 408
\inclinatum ... 298
\inclinatum ... 304
\include ... 327
\ionian ... 15
\italianChords ... 273
\italic ... 175, 478
\justified-lines ... 512
\justify ... 179, 490
\justify-field ... 490
\justify-string ... 491
\keepWithTag ... 330
\key ... 15, 28
\label ... 325
\laissezVibrer ... 37
\large ... 154, 177, 478
\larger ... 175, 177, 478
\layout ... 318, 352
\left-align ... 177, 491
\left-column ... 492
\line ... 492
\linea ... 298
\linea ... 304
\locrian ... 15
\longa ... 31, 39
\lookup ... 509
\lower ... 177, 492

\lydian ... 15
\lyricmode ... 191, 194
\lyricsto ... 194
\magnify ... 175, 479
\major ... 15
\makeClusters ... 112
\mark ... 77, 167
\markalphabet ... 509
\markletter ... 509
\markup ... 171, 173
\markuplines ... 172, 186
\maxima ... 31, 39
\medium ... 479
\melisma ... 198
\melismaEnd ... 198
\mergeDifferentlyDottedOff ... 116
\mergeDifferentlyDottedOn ... 116
\mergeDifferentlyHeadedOff ... 116
\mergeDifferentlyHeadedOn ... 116
\mf ... 86
\midi ... 318
\minor ... 15
\mixolydian ... 15
\mm ... 408
\mp ... 86
\musicglyph ... 502
\natural ... 502
\new ... 390
\noBreak ... 353
\noPageBreak ... 355
\noPageTurn ... 356
\normal-size-sub ... 479
\normal-size-super ... 479
\normal-text ... 480
\normalsize ... 154, 177, 480
\note ... 503
\note-by-number ... 503
\null ... 510
\number ... 480
\numericTimeSignature ... 46
\octaveCheck ... 8
\on-the-fly ... 510
\oneVoice ... 113
\open ... 220
\oriscus ... 298
\oriscus ... 304
\ottava ... 16
\override ... 403, 510
\override-lines ... 512
\p ... 86
\pad-around ... 181, 492
\pad-markup ... 181, 493
\pad-to-box ... 181, 493
\pad-x ... 181, 493
\page-ref ... 325, 510
\pageBreak ... 355
\pageTurn ... 356
\paper ... 318, 346
\parallelMusic ... 123
\parenthesize ... 160
\partcombine ... 120
\partial ... 48
\partial ... 102, 103
\pes ... 304
\phrasingSlurDashed ... 93
\phrasingSlurDotted ... 93
\phrasingSlurDown ... 93
\phrasingSlurNeutral ... 93
\phrasingSlurSolid ... 93
\phrasingSlurUp ... 93
\phrygian ... 15
\pitchedTrill ... 100
\postscript ... 182, 500
\pp ... 86
\ppp ... 86
\pppp ... 86
\predefinedFretboardsOff ... 246
\predefinedFretboardsOn ... 246
\property dentro de \lyricmode ... 191
\pt ... 408
\put-adjacent ... 494
\quilisma ... 298
\quilisma ... 304
\raise ... 177, 494
\relative ... 2, 4, 11, 210
\RemoveEmptyRhythmicStaffContext ... 140
\RemoveEmptyStaffContext ... 139
\removeWithTag ... 330
\repeat ... 102
\repeat percent ... 108
\repeat tremolo ... 110
\repeatTie ... 37, 103
\rest ... 39
\rfz ... 86
\right-align ... 177, 494
\right-column ... 494
\rightHandFinger ... 248
\roman ... 481
\rotate ... 495
\rounded-box ... 180, 501
\sacredHarpHeads ... 28
\sans ... 481
\scaleDurations ... 35
\score ... 316, 318, 503
\semiflat ... 504
\semiGermanChords ... 273
\semisharp ... 504
\sesquiflat ... 505
\sesquisharp ... 505
\set ... 401
\sf ... 86
\sff ... 86
\sfz ... 86
\sharp ... 505
\shiftOff ... 116
\shiftOn ... 116
\shiftOnn ... 116
\shiftOnnn ... 116
\showKeySignature ... 262
\showStaffSwitch ... 211
\simple ... 481
\skip ... 41
\slashed-digit ... 510
\slurDashed ... 92
\slurDotted ... 92
\slurDown ... 92
\slurNeutral ... 92
\slurSolid ... 92
\slurUp ... 93
\small ... 154, 177, 481

\smallCaps ... 482
\smaller ... 175, 177, 482
\sostenutoOff ... 214
\sostenutoOn ... 214
\sp ... 86
\spp ... 86
\startGroup ... 164
\startStaff ... 134
\startTrillSpan ... 100
\stemDown ... 160
\stemNeutral ... 160
\stemUp ... 160
\stencil ... 511
\stopGroup ... 164
\stopStaff ... 134
\stopTrillSpan ... 100
\storePredefinedDiagram ... 241
\stropha ... 298
\stropha ... 304
\strut ... 511
\sub ... 175, 482
\super ... 175, 483
\sustainOff ... 214
\sustainOn ... 214
\table-of-contents ... 327
\tag ... 330
\taor ... 262
\teeny ... 154, 177, 483
\tempo ... 142
\text ... 483
\textLengthOff ... 166
\textLengthOn ... 166
\thumb ... 156
\tied-lyric ... 505
\tieDashed ... 37
\tieDotted ... 37
\tieDown ... 37
\tieNeutral ... 37
\tieSolid ... 37
\tieUp ... 37
\time ... 46
\times ... 33
\tiny ... 154, 177, 483
\tocItem ... 327
\translate ... 178, 495
\translate-scaled ... 178, 495
\transparent ... 511
\transpose ... 4, 9, 11
\transposition ... 17
\treCorde ... 214
\triangle ... 182, 501
\trill ... 100
\tupletDown ... 33
\tupletNeutral ... 33
\tupletUp ... 33
\tweak ... 403
\typewriter ... 484
\unaCorda ... 214
\underline ... 175, 484
\unfoldRepeats ... 338
\unHideNotes ... 158
\unset ... 402
\upbow ... 220
\upright ... 484
\vcenter ... 496
\verbatim-file ... 511
\virga ... 298
\virga ... 304
\virgula ... 296
\voiceFourStyle ... 116
\voiceNeutralStyle ... 116
\voiceOne ... 113
\voiceOne ... \voiceFour ... 113
\voiceOneStyle ... 116
\voiceThreeStyle ... 116
\voiceTwoStyle ... 116
\whiteout ... 511
\with ... 391
\with-color ... 158, 511
\with-dimensions ... 512
\with-url ... 501
\wordwrap ... 179, 496
\wordwrap-field ... 496
\wordwrap-internal ... 512
\wordwrap-lines ... 512
\wordwrap-string ... 497
\wordwrap-string-internal ... 513

|

| ... 76

~

~ ... 36

A

add ChordShape ... 241
aeolian ... 15
after-title-space ... 346
aikenHeads ... 28
annotate-spacing ... 385
arpeggio ... 97
arpeggioArrowDown ... 97
arpeggioArrowUp ... 97
arpeggioBracket ... 98
arpeggioNormal ... 97
arpeggioParenthesis ... 98
arranger ... 320
aug ... 265
auto-first-page-number ... 348
autoBeaming ... 58
autoBeamSettings ... 58
autochange ... 210

B

Balloon_engraver ... 161
balloonGrobText ... 161
balloonLengthOff ... 161
balloonLengthOn ... 161
balloonText ... 161
banjo-c-tuning ... 251
banjo-modal-tuning ... 251
banjo-open-d-tuning ... 251
banjo-open-dm-tuning ... 251
barCheckSynchronize ... 76
barNumberVisibility ... 73

base-shortest-duration 374
before-title-space 346
bendAfter 95
between-system-padding 346
between-system-space 346
between-title-space 346
blank-last-page-force 348
blank-page-force 348
bookTitleMarkup 323
bottom-margin 346
bracket 90, 214
breakbefore 320
breathe 94

C

change 209
chordmode 4, 11, 238
chordNameExceptions 272
ChordNames 238
chordNameSeparator 272
chordNoteNamer 272
chordPrefixSpacer 273
chordRootNamer 272
clef 12
color 158
common-shortest-duration 374
composer 320
controlpitch 8
copyright 320
cr 86
crescHairpin 87
crescTextCresc 87
cross 27
cross-staff 212
currentBarNumber 73, 83

D

decr 86
dedicatoria 320
default 19, 20
defaultBarType 72
dim 265
dimHairpin 87
dimTextDecr 87
dimTextDecresc 87
dimTextDim 87
divisible 57
dodecaphonic 23
dorian 15
dynamic 90
dynamicDown 88
DynamicLineSpanner 87
dynamicNeutral 88
dynamicUp 88

E

easyHeadsOff 28
easyHeadsOn 28
espressivo 87
evenFooterMarkup 324
evenHeaderMarkup 323

F

f 86
ff 86
fff 86
ffff 86
finger 155
first-page-number 349
flag-style 212
followVoice 211
font-interface 155, 186
font-size 154, 155
fontSize 154
foot-separation 346
forget 24
four-string-banjo 251
fp 86
fret-diagram 231
fret-diagram-interface 236
fret-diagram-terse 232
fret-diagram-verbose 234
FretBoards 238

G

glissando 96
Grid_line_span_engraver 162
Grid_point_engraver 162
gridInterval 162

H

head-separation 346
hideKeySignature 262
hideNotes 158
hideStaffSwitch 211
horizontal-shift 348
Horizontal_bracket_engraver 164
huge 154

I

improvisationOff 30
improvisationOn 30
indent 348, 377
instrument 320
ionian 15

K

key 15, 28

L

large 154
layout file 351
left-margin 348
length 212
line-width 348, 377
locrian 15
ly:minimal-breaking 356
ly:optimal-breaking 355
ly:page-turn-breaking 355
lydian 15

M

m ... 265
magstep ... 154, 409
maj ... 265
major ... 15
major seven symbols ... 273
majorSevenSymbol ... 272
make-dynamic-script ... 91
make-pango-font-tree ... 188
makeClusters ... 112
measureLength ... 83
measurePosition ... 48
measurePosition ... 83
mergeDifferentlyDottedOff ... 116
mergeDifferentlyDottedOn ... 116
mergeDifferentlyHeadedOff ... 116
mergeDifferentlyHeadedOn ... 116
meter ... 320
mf ... 86
minimumFret ... 225
minimumPageTurnLength ... 356
minimumRepeatLengthForPageTurn ... 356
minor ... 15
mixed ... 214
mixolydian ... 15
modern ... 21
modern-cautionary ... 21
modern-voice ... 21
modern-voice-cautionary ... 22
mp ... 86

N

neo-modern ... 23
neo-modern-cautionary ... 23
no-reset ... 24
normalsize ... 154

O

octaveCheck ... 8
oddFooterMarkup ... 324
oddHeaderMarkup ... 323
oneVoice ... 113
opus ... 320
ottava ... 16
outside-staff-horizontal-padding ... 372
outside-staff-padding ... 372
outside-staff-priority ... 372

P

p ... 86
page-breaking-between-system-padding ... 349
page-count ... 349
page-limit-inter-system-space ... 349
page-limit-inter-system-space-factor ... 349
page-spacing-weight ... 349
page-top-space ... 346
paper-height ... 347
paper-width ... 348
parallelMusic ... 123
parenthesize ... 160
partcombine ... 120
pedalSustainStyle ... 214
percent ... 108
phrasingSlurDashed ... 93
phrasingSlurDotted ... 93
phrasingSlurDown ... 93
phrasingSlurNeutral ... 93
phrasingSlurSolid ... 93
phrasingSlurUp ... 93
phrygian ... 15
piano ... 22
piano-cautionary ... 22
PianoStaff ... 208, 210
piece ... 320
pipeSymbol ... 76
pitchedTrill ... 100
poet ... 320
pp ... 86
ppp ... 86
pppp ... 86
predefinedFretboardsOff ... 246
predefinedFretboardsOn ... 246
print-all-headers ... 323, 349
print-first-page-number ... 349
print-page-number ... 349

Q

quotedEventTypes ... 150

R

r ... 39
R ... 42
ragged-bottom ... 349
ragged-last ... 349, 377
ragged-last-bottom ... 349
ragged-right ... 350, 377
relative ... 2, 4, 11, 210
repeatCommands ... 105
rfz ... 86
rgb-color ... 159
rightHandFinger ... 248

S

s ... 41
sacredHarpHeads ... 28
scoreTitleMarkup ... 323
set-accidental-style ... 19
set-octavation ... 16
sf ... 86
sff ... 86
sfz ... 86
shiftOff ... 116
shiftOn ... 116
shiftOnn ... 116
shiftOnnn ... 116
short-indent ... 348
show-available-fonts ... 188
showFirstLength ... 334
showKeySignature ... 262
showLastLength ... 334
showStaffSwitch ... 211
skipTypesetting ... 334

slurDashed ... 92
slurDotted ... 92
slurDown ... 92
slurNeutral ... 92
slurSolid ... 92
slurUp ... 93
small ... 154
sostenutoOff ... 214
sostenutoOn ... 214
sp ... 86
spacing ... 374
spp ... 86
staff-padding ... 209
Staff.midiInstrument ... 335
start-repeat ... 105
startGroup ... 164
startTrillSpan ... 100
Stem ... 212
stem-spacing-correction ... 374
stemDown ... 160
stemLeftBeamCount ... 67
stemNeutral ... 160
stemRightBeamCount ... 67
stemUp ... 160
stopGroup ... 164
stopTrillSpan ... 100
storePredefinedDiagram ... 241
stringTunings ... 238
StringTunings ... 228
subdivideBeams ... 61
subsubtitle ... 320
subtitle ... 320
suggestAccidentals ... 292
sus ... 268
sustainOff ... 214
sustainOn ... 214
system-count ... 350
system-separator-markup ... 350

T

TabStaff ... 225
TabVoice ... 225
tagline ... 321
taor ... 262
teaching ... 23
teeny ... 154
text ... 214
textSpannerDown ... 167
textSpannerNeutral ... 167
textSpannerUp ... 167
thumb ... 156
tiny ... 154
title ... 320
top-margin ... 347
transpose ... 4, 9, 11
transposition ... 17
treCorde ... 214
tremolo ... 110
tremoloFlags ... 110
trill ... 100
TupletNumber ... 34
tupletNumberFormatFunction ... 33
tupletSpannerDuration ... 33

U

unaCorda ... 214
unfold ... 107
unHideNotes ... 158

V

voice ... 19, 20
Voice ... 113
voiceOne ... 113

W

whichBar ... 72
with-color ... 158

X

x11-color ... 158, 159

Apéndice F Índice de LilyPond

Además de todas las instrucciones y palabras clave de LilyPond, este índice es una lista de términos musicales y las palabras que tienen relación con cada uno de ellos, con enlaces a aquellas secciones del manual que describen o se ocupan de dicho término. Cada uno de los enlaces consta de dos partes. La primera parte apunta a la situación exacta del manual en que aparece el término; la segunda parte apunta al comienzo de la sección correspondiente del manual en la que se discute dicho término.

!

! 5

'

' 1

(

(begin * * * *) 58
(end * * * *) 58

,

, ... 1

-

- 84

.

. 32

/

/ 268
/+ 268

:

: 110

<

< 111
<...> 111

=

= 8

>

> 111

?

? 5

[

[..... 67

]

] 67

^

^ 268

_

_ 192, 196

\

\! 86
\(..... 93
\) 93
\< 86
\> 86
\abs-fontsize 476
\accepts 395, 396
\addChordShape 241
\addlyrics 193
\addlyrics 194
\aeolian 15
\afterGrace 79
\aikenHeads 28
\allowPageTurn 356
\alternative 102
\AncientRemoveEmptyStaffContext 140
\applyContext 447
\applyOutput 447
\arpeggio 97, 98
\arpeggioArrowDown 97, 98
\arpeggioArrowUp 97, 98
\arpeggioBracket 98
\arpeggioNormal 97, 98
\arpeggioParenthesis 98
\arrow-head 182, 498
\ascendens 298
\ascendens 304
\auctum 298
\auctum 304
\augmentum 304
\autoBeamOff 67
\autoBeamOn 67
\autochange 210
\backslashed-digit 508
\balloonGrobText 161

\balloonLengthOff ... 161
\balloonLengthOn ... 161
\balloonText ... 161
\bar ... 70
\beam ... 498
\bendAfter ... 95
\bold ... 175, 476
\book ... 317, 318
\bookpart ... 318, 354
\box ... 180, 476
\bracket ... 90, 180, 498
\break ... 353
\breathe ... 94
\breve ... 31, 39
\cadenzaOff ... 49
\cadenzaOn ... 49
\caesura ... 296
\caps ... 477
\cavum ... 298
\cavum ... 304
\center-align ... 177, 485
\center-column ... 179, 485
\change ... 209
\char ... 508
\chordmode ... 4, 11, 238
\chords ... 270
\circle ... 180, 498
\clef ... 12
\cm ... 408
\column ... 179, 485
\column-lines ... 512
\combine ... 182, 486
\compressFullBarRests ... 42
\concat ... 486
\context ... 390
\cr ... 86
\crescHairpin ... 87
\crescTextCresc ... 87
\decr ... 86
\defaultTimeSignature ... 46
\deminutum ... 298
\deminutum ... 304
\denies ... 395, 396
\descendens ... 298
\descendens ... 304
\dimHairpin ... 87
\dimTextDecr ... 87
\dimTextDecresc ... 87
\dimTextDim ... 87
\dir-column ... 486
\displayLilyMusic ... 333, 439
\displayMusic ... 438
\divisioMaior ... 296
\divisioMaxima ... 296
\divisioMinima ... 296
\dorian ... 15
\dotsDown ... 32
\dotsNeutral ... 32
\dotsUp ... 32
\doubleflat ... 502
\doublesharp ... 502
\downbow ... 220
\draw-circle ... 182, 499
\draw-line ... 182, 499
\dynamic ... 90, 477
\dynamicDown ... 88
\dynamicNeutral ... 88
\dynamicUp ... 88
\easyHeadsOff ... 28
\easyHeadsOn ... 28
\epsfile ... 182, 499
\espressivo ... 87
\expandFullBarRests ... 42
\f ... 86
\featherDurations ... 69
\ff ... 86
\fff ... 86
\ffff ... 86
\fill-line ... 179, 487
\filled-box ... 182, 499
\finalis ... 296
\finger ... 155, 477
\flat ... 502
\flexa ... 304
\fontCaps ... 477
\fontsize ... 175, 477
\fp ... 86
\fraction ... 508
\frenchChords ... 273
\fret-diagram ... 231, 505
\fret-diagram-terse ... 232, 506
\fret-diagram-verbose ... 234, 506
\fromproperty ... 508
\general-align ... 178, 487
\germanChords ... 273
\glissando ... 96
\grace ... 78
\halign ... 177, 488
\harmonic ... 221
\harp-pedal ... 507
\hbracket ... 180, 500
\hcenter-in ... 489
\header ... 318
\hideKeySignature ... 262
\hideNotes ... 158
\hideStaffSwitch ... 211
\hspace ... 489
\huge ... 154, 177, 478
\improvisationOff ... 30
\improvisationOn ... 30
\in ... 408
\inclinatum ... 298
\inclinatum ... 304
\include ... 327
\ionian ... 15
\italianChords ... 273
\italic ... 175, 478
\justified-lines ... 512
\justify ... 179, 490
\justify-field ... 490
\justify-string ... 491
\keepWithTag ... 330
\key ... 15, 28
\label ... 325
\laissezVibrer ... 37
\large ... 154, 177, 478
\larger ... 175, 177, 478
\layout ... 318, 352
\left-align ... 177, 491
\left-column ... 492

\line ... 492
\linea ... 298
\linea ... 304
\locrian ... 15
\longa ... 31, 39
\lookup ... 509
\lower ... 177, 492
\lydian ... 15
\lyricmode ... 191, 194
\lyricsto ... 194
\magnify ... 175, 479
\major ... 15
\makeClusters ... 112
\mark ... 77, 167
\markalphabet ... 509
\markletter ... 509
\markup ... 171, 173
\markuplines ... 172, 186
\maxima ... 31, 39
\medium ... 479
\melisma ... 198
\melismaEnd ... 198
\mergeDifferentlyDottedOff ... 116
\mergeDifferentlyDottedOn ... 116
\mergeDifferentlyHeadedOff ... 116
\mergeDifferentlyHeadedOn ... 116
\mf ... 86
\midi ... 318
\minor ... 15
\mixolydian ... 15
\mm ... 408
\mp ... 86
\musicglyph ... 502
\natural ... 502
\new ... 390
\noBreak ... 353
\noPageBreak ... 355
\noPageTurn ... 356
\normal-size-sub ... 479
\normal-size-super ... 479
\normal-text ... 480
\normalsize ... 154, 177, 480
\note ... 503
\note-by-number ... 503
\null ... 510
\number ... 480
\numericTimeSignature ... 46
\octaveCheck ... 8
\on-the-fly ... 510
\once ... 402
\oneVoice ... 113
\open ... 220
\oriscus ... 298
\oriscus ... 304
\ottava ... 16
\override ... 403, 510
\override-lines ... 512
\p ... 86
\pad-around ... 181, 492
\pad-markup ... 181, 493
\pad-to-box ... 181, 493
\pad-x ... 181, 493
\page-ref ... 325, 510
\pageBreak ... 355
\pageTurn ... 356
\paper ... 318
\paper ... 324
\paper ... 346
\parallelMusic ... 123
\parenthesize ... 160
\partcombine ... 120
\partial ... 48
\partial ... 102, 103
\pes ... 304
\phrasingSlurDashed ... 93
\phrasingSlurDotted ... 93
\phrasingSlurDown ... 93
\phrasingSlurNeutral ... 93
\phrasingSlurSolid ... 93
\phrasingSlurUp ... 93
\phrygian ... 15
\pitchedTrill ... 100
\postscript ... 182, 500
\pp ... 86
\ppp ... 86
\pppp ... 86
\predefinedFretboardsOff ... 246
\predefinedFretboardsOn ... 246
\property dentro de \lyricmode ... 191
\pt ... 408
\put-adjacent ... 494
\quilisma ... 298
\quilisma ... 304
\raise ... 177, 494
\relative ... 2, 4, 11, 210
\RemoveEmptyRhythmicStaffContext ... 140
\RemoveEmptyStaffContext ... 139
\removeWithTag ... 330
\repeat ... 102
\repeat percent ... 108
\repeat tremolo ... 110
\repeatTie ... 37, 103
\rest ... 39
\rfz ... 86
\right-align ... 177, 494
\right-column ... 494
\rightHandFinger ... 248
\roman ... 481
\rotate ... 495
\rounded-box ... 180, 501
\sacredHarpHeads ... 28
\sans ... 481
\scaleDurations ... 35
\score ... 316, 318, 503
\semiflat ... 504
\semiGermanChords ... 273
\semisharp ... 504
\sesquiflat ... 505
\sesquisharp ... 505
\set ... 401
\sf ... 86
\sff ... 86
\sfz ... 86
\sharp ... 505
\shiftOff ... 116
\shiftOn ... 116
\shiftOnn ... 116
\shiftOnnn ... 116
\showKeySignature ... 262
\showStaffSwitch ... 211

\simple ... 481
\skip ... 41
\slashed-digit ... 510
\slurDashed ... 92
\slurDotted ... 92
\slurDown ... 92
\slurNeutral ... 92
\slurSolid ... 92
\slurUp ... 93
\small ... 154, 177, 481
\smallCaps ... 482
\smaller ... 175, 177, 482
\sostenutoOff ... 214
\sostenutoOn ... 214
\sp ... 86
\spp ... 86
\startGroup ... 164
\startStaff ... 134
\startTrillSpan ... 100
\stemDown ... 160
\stemNeutral ... 160
\stemUp ... 160
\stencil ... 511
\stopGroup ... 164
\stopStaff ... 134
\stopTrillSpan ... 100
\storePredefinedDiagram ... 241
\stropha ... 298
\stropha ... 304
\strut ... 511
\sub ... 175, 482
\super ... 175, 483
\sustainOff ... 214
\sustainOn ... 214
\table-of-contents ... 327
\tag ... 330
\taor ... 262
\teeny ... 154, 177, 483
\tempo ... 142
\text ... 483
\textLengthOff ... 166
\textLengthOn ... 166
\thumb ... 156
\tied-lyric ... 505
\tieDashed ... 37
\tieDotted ... 37
\tieDown ... 37
\tieNeutral ... 37
\tieSolid ... 37
\tieUp ... 37
\time ... 46
\times ... 33
\tiny ... 154, 177, 483
\tocItem ... 327
\translate ... 178, 495
\translate-scaled ... 178, 495
\transparent ... 511
\transpose ... 4, 9, 11
\transposition ... 17
\treCorde ... 214
\triangle ... 182, 501
\trill ... 100
\tupletDown ... 33
\tupletNeutral ... 33
\tupletUp ... 33
\tweak ... 403
\typewriter ... 484
\unaCorda ... 214
\underline ... 175, 484
\unfoldRepeats ... 338
\unHideNotes ... 158
\unset ... 402
\upbow ... 220
\upright ... 484
\vcenter ... 496
\verbatim-file ... 511
\virga ... 298
\virga ... 304
\virgula ... 296
\voiceFourStyle ... 116
\voiceNeutralStyle ... 116
\voiceOne ... 113
\voiceOne ... \voiceFour ... 113
\voiceOneStyle ... 116
\voiceThreeStyle ... 116
\voiceTwoStyle ... 116
\whiteout ... 511
\with ... 391
\with-color ... 158, 511
\with-dimensions ... 512
\with-url ... 501
\wordwrap ... 179, 496
\wordwrap-field ... 496
\wordwrap-internal ... 512
\wordwrap-lines ... 512
\wordwrap-string ... 497
\wordwrap-string-internal ... 513

|

| ... 76

~

~ ... 36

1

15ma ... 16

8

8va ... 16

A

a due ... 123
a due, partes ... 120
abajo, arco ... 220
absoluta ... 1
absoluta, especificación, de la octava ... 1
absoluta, introducción, de la octava ... 1
absolutos, matices ... 86
accent ... 513
acciaccatura ... 78
acciaccatura ... 81, 432, 538
accidental ... 4
Accidental ... 6, 25, 291, 296
accidental-interface ... 6

accidental-suggestion-interface ... 25
Accidental_engraver ... 6, 25, 292
AccidentalCautionary ... 6
AccidentalPlacement ... 25
accidentals ... 296
Accidentals and key signatures ... 4, 6, 15
Accidentals and key signatures ... 16
Accidentals and key signatures ... 312
AccidentalSuggestion ... 25, 292
acento ... 85
acento ... 86
acento ... 513
acorde ... 112
acorde ... 264, 271
acorde arpegiado ... 97
acorde quebrado ... 97
acordeón ... 215
acordeón, cambios en el ... 215
acordeón, signos de cambio en el ... 215
acordeón, signos de discanto ... 215
acordes ... 111, 269
acordes alterados ... 266
acordes de jazz ... 264, 272
acordes de pentagrama cruzado ... 212
acordes e introducción relativa de la octava ... 3
acordes extendidos ... 266
acordes, alteración de notas ... 268
acordes, alteraciones dentro de ... 25
acordes, bajo de ... 268
acordes, diagramas automáticos de ... 245
acordes, diagramas de ... 229
acordes, diagramas de trastes con nombres de ... 238
acordes, digitación de ... 156
acordes, división entre pentagramas con \autochange ... 211
acordes, formas de, en instrumentos de trastes ... 241
acordes, inversiones ... 268
acordes, modo de ... 264
acordes, nombres de ... 264, 269
acordes, nombres de, en MIDI ... 338
acordes, supresiones en ... 268
acordes, variantes de ... 265
add ChordShape ... 241
addChordShape ... 433, 539
adding a white background to text ... 511
Adding and removing engravers ... 53
addInstrumentDefinition ... 433, 539
addQuote ... 433, 539
adicionales, establecer líneas ... 133
adiciones en acordes ... 267
adorno, notas de, dentro de corchetes de grupo especial ... 35
adorno, seguimiento de notas de ... 79
aeolian ... 15
afinaciones del banjo ... 251
after-title-space ... 346
afterGrace ... 433, 539
agrupación de pulsos ... 61
Aiken, cabezas de nota con forma de ... 28
aikenHeads ... 28
ajuste de línea, texto con ... 179
ajuste de líneas adicionales ... 133
ajuste del símbolo del pentagrama ... 133, 409
ajuste fino de las barras automáticas ... 58
al aire, indicación de cuerda ... 220
al niente ... 88
al niente ... 90
alignAboveContext ... 395
alignBelowContext ... 395
alineación a objetos ... 424
alineación a una cadenza (cadencia) ... 82
alinear texto ... 177
All layout objects ... 401, 421
allowPageTurn ... 433, 539
alteración con paréntesis ... 5
alteración de cuartos de tono ... 6
alteración de precaución ... 5
alteración de voz de precaución moderno, estilo de ... 22
alteración en nota ligada ... 5
alteración forzada para trino con nota ... 101
alteración recordatoria ... 5
alteración, estilo de voz de ... 20
alteración, estilo de voz de precaución moderno ... 22
alteración, estilo de, olvidar ... 24
alteración, estilo moderno de ... 21
alteración, moderno de precaución, voz, estilo de ... 22
alteraciones ... 291
alteraciones accidentales ... 19, 292
alteraciones automáticas ... 19
alteraciones de la musica ficta ... 292
alteraciones de piano ... 22
alteraciones en acordes ... 25
alteraciones estilo moderno ... 20
alteraciones modernas ... 21
alteraciones no restablecer, estilo de ... 24
alteraciones y notas simultáneas ... 25
alteraciones, en varias voces ... 21
alteraciones, estilo de las ... 19
alteraciones, estilo de, neo-moderno ... 23
alteraciones, estilo moderno ... 21
alteraciones, estilo moderno de precaución ... 21
alteraciones, estilo predeterminado ... 20
alteraciones, estilo predeterminado de las ... 19
alteraciones, estilo teaching (enseñanza) ... 23
alteraciones, estilo, dodecafónico ... 23
alteraciones, estilo, moderno-precaución ... 20
alteraciones, estilo, neo-moderno de precaución ... 23
alteraciones, piano de precaución ... 22
alteraciones, piano de precaución, estilo de ... 22
alternativos, finales, en repeticiones explícitas ... 107
alto, clave de ... 12
altura de concierto ... 18
alturas ... 1
alturas en MIDI ... 338
alturas, nombres de, otros idiomas ... 6
alturas, transporte de ... 9
Amazing Grace, ejemplo de gaita ... 262
ambitus ... 25
Ambitus ... 27
ambitus-interface ... 27
Ambitus_engraver ... 27
AmbitusAccidental ... 27
AmbitusLine ... 27
AmbitusNoteHead ... 27
anacrusa ... 48
anacrusa ... 49
anacrusa en repeticiones ... 103
análisis musicológico ... 164
ángulo, paréntesis en ... 111

ángulo, reguladores en ... 420
anidada, repetición ... 105
anidados, corchetes de pentagrama ... 131
anidamiento de pentagramas ... 131
annotate-spacing ... 385
antiguas, claves ... 12
apagado ... 513
apaisado ... 345
applyContext ... 433, 539
applyMusic ... 433, 539
applyOutput ... 433, 539
appoggiatura ... 78
appoggiatura ... 81, 433, 539
árabe, ejemplo de música ... 314
árabe, música ... 309
árabe, plantilla de música ... 314
árabes, armaduras ... 312
Arabic key signatures ... 311
Arabic note names ... 310
Arabic semi-flat symbol ... 310
arco abajo ... 84, 513
arco abajo, indicación ... 220
arco arriba ... 84, 513
arco arriba, indicación ... 220
arco, indicaciones de ... 220
armadura ... 291
Armadura de la tonalidad ... 15
armadura, visibilidad después de un cambio explícito ... 418
armónicos ... 222
armónicos artificiales ... 221
armónicos en instrumentos de trastes ... 250
armónicos en tablatura ... 226
armónicos naturales ... 221
armónicos, cabezas de nota ... 27
arpa sacra, cabezas de nota de ... 28
arpa, diagramas de pedal ... 219
arpa, pedales ... 219
arpeggio ... 97
Arpeggio ... 100
arpeggioArrowDown ... 97
arpeggioArrowUp ... 97
arpeggioBracket ... 98
arpeggioNormal ... 97
arpeggioParenthesis ... 98
arpegiado, acorde, símbolos especiales ... 98
arpegio ... 97
arpegio ... 100
arpegio con paréntesis de pentagrama cruzado ... 100
arpegio de pentagrama cruzado con paréntesis ... 100
arranger ... 320
arriba, arco ... 220
articulaciones ... 84, 297
Articulation and dynamics ... 90
articulation-event ... 150
artificiales, armónicos ... 221
aspas, cabezas de nota ... 27
assertBeamQuant ... 433, 539
assertBeamSlope ... 433, 539
aug ... 265
auto-first-page-number ... 348
autoBeaming ... 58
autoBeamOff ... 56
autoBeamOn ... 56
autoBeamSettings ... 58
autocambiador y el modo relativo ... 210
autochange ... 210
autochange ... 433, 539
AutoChangeMusic ... 211
automática, combinación de partes ... 120
automáticas, alteraciones ... 19
automáticas, generación de barras ... 58
automaticBars ... 419
automático, cambio de pentagrama ... 210
automáticos, diagramas de acordes ... 245
automáticos, diagramas de traste ... 245
Axis_group_engraver ... 362
ayuda, globos de ... 161

B

Backend ... 397, 401
backslashed digits ... 508
bajo cifrado ... 277
bajo cifrado, alineación ... 281
bajo cifrado, líneas de extensión ... 279
bajo de los acordes ... 268
bajo, clave de ... 12
balloon-interface ... 162
Balloon_engraver ... 161, 162
balloonGrobText ... 161
balloonGrobText ... 433, 539
balloonLengthOff ... 161
balloonLengthOn ... 161
balloonText ... 161
balloonText ... 433, 539
BalloonTextItem ... 162
banjo, afinaciones del ... 251
banjo, tablatura de ... 223
banjo, tablaturas de ... 250
banjo-c-tuning ... 251
banjo-modal-tuning ... 251
banjo-open-d-tuning ... 251
banjo-open-dm-tuning ... 251
Banter ... 272
bar ... 433, 539
bar number alignment ... 74
bar numbers, regular spacing ... 73
bar-line-interface ... 525
Bar_engraver ... 271
barCheckSynchronize ... 76
barítono, clave de ... 12
BarLine ... 72
BarNumber ... 75
barNumberCheck ... 433, 539
barNumberVisibility ... 73
barradas, cabezas de nota ... 30
barras automáticas ... 58
barras automáticas, ajuste fino ... 58
barras de compás ... 70
barras de pentagrama cruzado ... 209
barras de trémolo ... 110
barras manuales ... 56, 67
barras progresivas ... 69
Bartók pizzicato ... 222
base-shortest-duration ... 374
Bass, figured ... 276
Bass, thorough ... 276
BassFigure ... 280, 281

BassFigureAlignment ... 280, 281
BassFigureBracket ... 280, 281
BassFigureContinuation ... 280, 281
BassFigureLine ... 280, 281
Basso continuo ... 276
batería ... 251, 253
Beam ... 58, 210, 227
becuadro ... 4
before-title-space ... 346
bemol ... 4
bemol ... 6
bemol, doble ... 4
bendAfter ... 95
bendAfter ... 433, 539
between-system-padding ... 346
between-system-space ... 346
between-title-space ... 346
bisbiglando ... 218
blank-last-page-force ... 348
blank-page-force ... 348
bloque MIDI ... 337
bookTitleMarkup ... 323
bottom-margin ... 346
bracket ... 90, 214
break-align-symbols ... 424
break-visibility ... 416
breakbefore ... 320
breathe ... 94
breathe ... 433, 539
BreathingSign ... 95, 298
breve ... 32
breve ... 39
buscar objetos gráficos ... 403

C

cabezas de nota antiguas ... 289
cadencia ... 49
cadenza ... 50, 83
cadenza (cadencia) ... 82
cadenza (cadencia), alinear a ... 82
caesura ... 95
caída ... 96
caídas de tono (falls) ... 95
calderón ... 84, 513
calderón sobre la línea divisoria ... 167
calderón sobre un silencio multicompás ... 43
cambbio manual de pentagrama ... 209
cambiar propiedades ... 401
cambio automático de pentagrama ... 210
cambio de dedo ... 155
cambio de instrumento ... 147
cantante, nombre del ... 203
capas ... 415
caracteres especiales en modo de marcado ... 173
cejilla, indicación de ... 230
centering a column of text ... 485
centrar indicaciones dinámicas en música para piano ... 209
centrar texto en la página ... 179
cesura ... 95
change ... 209
changing direction of text columns ... 486
ChoirStaff ... 131
ChoirStaff ... 132
chord diagrams ... 238
chord, modifying one note in ... 404
Chord_name_engraver ... 271
chordmode ... 4, 11, 238
ChordName ... 271
chordNameExceptions ... 272
ChordNames ... 142, 238, 271
chordNameSeparator ... 272
chordNoteNamer ... 272
chordPrefixSpacer ... 273
chordRootNamer ... 272
chords and ties ... 37
cifrado americano ... 264
circling text ... 498
citar otras voces ... 148, 151
clave ... 12
clave antigua ... 12
clave de alto ... 12
clave de bajo ... 12
clave de barítono ... 12
clave de contrabajo ... 12
clave de Do ... 12
clave de Fa ... 12
clave de mezzosoprano ... 12
clave de Sol ... 12
clave de soprano ... 12
clave de tenor ... 12
clave de violín ... 12
clave francesa ... 12
clave transpositora ... 12
clave, visibilidad después de un cambio explícito ... 418
claves, visibilidad de la octavación ... 419
clef ... 4
clef ... 12
clef ... 433, 539
Clef ... 14
clef-interface ... 14
Clef_engraver ... 14
clefs ... 287, 295
cluster ... 112
Cluster_spanner_engraver ... 113
ClusterSpanner ... 113
ClusterSpannerBeacon ... 113
coda ... 78, 84, 513
coda sobre la línea divisoria ... 167
código, llamadas durante la interpretación ... 447
código, llamar sobre objetos de presentación ... 447
colisiones ... 116
Collisions of objects ... 119
colocación de los silencios multicompás ... 44
color ... 158
color en acordes ... 159
color rgb ... 159
coloreadas, notas ... 158
coloreadas, notas de acorde ... 159
coloreado de objetos ... 415
coloreados, objetos ... 158
colorear notas ... 158
colorear objetos ... 158
colores ... 158
colores de x11 ... 158
colores, lista de ... 460
coloring text ... 511

coloring voices ... 116
columnas, texto en ... 179
`Combining notes into chords` ... 112
comienzo de repetición ... 105
comienzo del sistema ... 127
comillas, en la letra ... 191, 192
`common-shortest-duration` ... 374
compás completo, silencios de ... 42
compás compuesto, indicaciones de ... 47
compás parcial ... 48
`compás polimétrico` ... 52
compás, barras invisibles de ... 71
compás, cambiar la longitud del ... 48
compás, comprobación de ... 76
compás, estilo de la indicación de ... 46
compás, indicación de ... 288
compás, número de ... 83
compás, número de, formato ... 74
compás, números de ... 73
compás, repetición de ... 108
compás, repeticiones y números de ... 105
compás, visibilidad de la indicación de ... 46
compases distintos al mismo tiempo ... 392
compases, agrupaciones de ... 61
compases, subgrupos dentro de ... 61
`Completion_heads_engraver` ... 53
`composer` ... 320
compresión de música ... 35
compuesto, indicaciones de compás ... 47
concatenating text ... 486
condensar silencios ... 45
`ContextChange` ... 210
Contexto, creación de ... 390
`Contexts` ... 388
`Contexts and engravers` ... 115, 388
contexts, nested ... 396
continua, ligadura ... 92
contrabajo, clave de ... 12
control, alturas de ... 8
control, trucar puntos de ... 405
controlling general text alignment ... 487
`controlpitch` ... 8
`copyright` ... 320
copyright ... 323
coral, clave de tenor ... 12
coral, partitura ... 198
corchete de casilla de repetición con texto ... 106
corchete de primera y segunda vez ... 105
corchete horizontal ... 164
corchete vertical ... 127
corchetes ... 164
corchetes anidados ... 131
corchetes de fraseo ... 164
coro, sistema de ... 127
`cr` ... 86
crear contextos ... 391
creating empty text objects ... 510
creating horizontal spaces in text ... 489
creating text fractions ... 508
creating vertical spaces in text ... 511
crescendo ... 86
`crescendo` ... 90
`crescHairpin` ... 87
`crescTextCresc` ... 87
`cross` ... 27
`cross-staff` ... 212
cruce de pentagramas ... 211
cruzado, acordes de pentagrama ... 212
cruzado, línea de pentagrama ... 211
cruzado, notas de pentagrama ... 212
cruzado, pentagrama ... 211
cruzado, pentagrama, barras de ... 209
cruzado, pentagrama, notas de ... 209
cruzado, plicas de pentagrama ... 212
cuartos de tono ... 4
cuartos de tono en MIDI ... 338
cuartos de tono, alteraciones ... 6
`cueDuring` ... 433, 539
cuerda al aire, indicación de ... 220
cuerda, número de la ... 224
cuerdas orquestales ... 220
cuerdas, escribir música para ... 220
`CueVoice` ... 154
`currentBarNumber` ... 73, 83
custodes ... 285
custos ... 285
`Custos` ... 286
`Custos_engraver` ... 286

D

D.S al Fine ... 78
decorar texto ... 180
`decr` ... 86
decrescendo ... 86
`decrescendo` ... 90
`dedicatoria` ... 320
`default` ... 19, 20
`Default_bar_line_engraver` ... 52
`defaultBarType` ... 72
derecha, mano, digitaciones para instrumentos de traste ... 248
derecho, pedal ... 214
desplazamiento ... 41
desplazamiento de silencios, automático ... 116
desplazamiento de voces ... 116
desplazamiento, silencio automático de ... 116
desplazar nota ... 116
desplegar música ... 107
desplegar repetición ... 107
detener líneas de pentagrama ... 133
detener un pentagrama ... 134
diagrama de trastes personalizado ... 236
diagramas de acordes para instrumentos de trastes ... 229
diagramas de traste, transporte de los ... 239
diagramas de trastes ... 229
diagramas de trastes personalizados, añadir ... 240
dibujar el símbolo del pentagrama ... 409
dibujar objetos gráficos ... 180
dibujo del símbolo del pentagrama ... 133
digitación ... 155
digitación de acordes ... 156
digitación frente a números de cuerda ... 224
digitación, instrucciones de, para acordes ... 156
digitaciones de la mano derecha para instrumentos de trastes ... 248
digitaciones, añadir, a diagramas de trastes ... 247
`dim` ... 265

dimHairpin ... 87
diminuendo ... 86
dimTextDecr ... 87
dimTextDecresc ... 87
dimTextDim ... 87
dinámica ... 86
dinámica, indicaciones de, centradas para música de teclado ... 209
dinámica, posición vertical ... 87
dinámica, varias indicaciones sobre una nota ... 87
dinámicas indicaciones editoriales ... 90
dinámicas, indicaciones entre paréntesis ... 90
dinámicas, indicaciones, nuevas ... 90
discanto, signos de, del acordeón ... 215
discontinua, ligadura ... 92
diseño «mensurstriche» ... 130
diseño de página ... 377
dispatcher ... 542
displayLilyMusic ... 433, 539
displayMusic ... 434, 540
disposición de la página ... 324
distances, absolute ... 408
distances, scaled ... 408
distancia entre pentagramas ... 360
distancia entre pentagramas en música de piano ... 212
divisible ... 57
divisio ... 296
división de notas ... 53
divisiones ... 296
Do, clave de ... 12
doble bemol ... 4
doble bemol ... 6
doble sostenido ... 4
doble sostenido ... 6
documentación interna ... 403
dodecafónico, estilo de alteraciones ... 23
dodecaphonic ... 23
doits (elevaciones de tono) ... 95
dorian ... 15
dórico, modo ... 15
DotColumn ... 32
Dots ... 32
double time signatures ... 50
DoublePercentRepeat ... 109
DoublePercentRepeatCounter ... 109
downbow ... 513
drawing beams within text ... 498
drawing boxes with rounded corners ... 499
drawing boxes with rounded corners around text ... 501
drawing circles within text ... 499
drawing lines within text ... 499
drawing solid boxes within text ... 499
drawing triangles within text ... 501
DrumStaff ... 127, 253, 259
DrumVoice ... 253, 254, 259
duración de las notas ... 31
duraciones automáticas de las sílabas ... 195
duraciones en MIDI ... 338
durations, scaling ... 35
dynamic ... 90
dynamic-event ... 150
dynamicDown ... 88
DynamicLineSpanner ... 87
DynamicLineSpanner ... 88, 90
dynamicNeutral ... 88
DynamicText ... 90
dynamicUp ... 88

E

easyHeadsOff ... 28
easyHeadsOn ... 28
eclesiásticos, silencios ... 44
editoriales, indicaciones dinámicas ... 90
elevación ... 96
elevaciones de tono (doits) ... 95
encabezamiento ... 324
enclosing text in a box with rounded corners ... 501
enclosing text within a box ... 476
endSpanners ... 434, 540
Engravers and Performers ... 388
Engravers explained ... 53
Engraving ... 374
enmarcar texto ... 180
ensayo, letra de, estilo ... 77
ensayo, letra de, formato ... 77
ensayo, letras de ... 77
eolio, modo ... 15
eonio, modo ... 15
escalado de pentagramas ... 135
escribir música en paralelo ... 123
Espaciado de la letra ... 201
espaciado de líneas adicionales ... 133
espaciado horizontal ... 373
espaciado vertical ... 360
espaciado, presentación del ... 385
espacio dentro de los sistemas ... 360
espacio entre pentagramas ... 360
espacios, en la letra ... 191, 192
especial, símbolos de arpegio ... 98
especiales, cabezas de figura ... 27
espressivo ... 84
espressivo ... 87
espressivo ... 513
espressivo, articulación ... 87
Establecer el comportamiento de las barras automáticas ... 516
estándar, tamaño de la tipografía (de notación) ... 155
estilo de la letra de ensayo ... 77
estilo de la ligadura ... 92
estilos de cabeza de nota ... 27
etiqueta ... 330
evenFooterMarkup ... 324
evenHeaderMarkup ... 323
excepciones, nombres de acorde ... 273
explícitas, repeticiones ... 107
explicitClefVisibility ... 418
explicitKeySignatureVisibility ... 418
Explicitly instantiating voices ... 115
extensora, línea ... 198

F

f ... 86
Fa, clave de ... 12
fácil, notación, cabezas de nota de ... 28
falls (caídas de tono) ... 95

fantasma, notas ... 160
FDL, GNU Free Documentation License ... 566
featherDurations ... 434, 540
fermata ... 513
Feta, tipografía ... 461
ff ... 86
fff ... 86
ffff ... 86
figura ... 32
figura, cabeza de la ... 27
Figured bass ... 276
FiguredBass ... 142, 280, 281
final de repetición ... 105
finales alternativos ... 102
finalis ... 296
finger ... 155
finger-interface ... 398
Fingering ... 157, 225, 397, 398
fingering-event ... 157, 397
Fingering_engraver ... 157, 397, 400
FingeringEvent ... 157, 397
first-page-number ... 349
Fixing overlapping notation ... 210
flag-style ... 212
flageolet ... 84, 513
flags ... 290
followVoice ... 211
font-interface ... 155, 186, 398, 510
font-size ... 154, 155
fontSize ... 154
foot marks ... 513
foot-separation ... 346
Forbid_line_break_engraver ... 53
forget ... 24
formas, notas con ... 28
Formatear el texto ... 533, 537
formato de la letra de ensayo ... 77
forte, pedal ... 214
four bar music ... 353
four-string-banjo ... 251
fp ... 86
fragmentos ... 148, 151
francesa, clave ... 12
francesa, partituras a la ... 139
francesa, pentagrama a la ... 139
francesa, pentagramas a la ... 135
fraseo, corchetes de ... 164
fraseo, en la letra ... 197
fraseo, ligadura de ... 92
fraseo, ligaduras de ... 93
fraseo, marcas de ... 93
fret diagrams ... 238
fret-diagram ... 231
fret-diagram, marcado con ... 231
fret-diagram-interface ... 236, 238, 242, 245, 248
fret-diagram-terse ... 232
fret-diagram-terse, marcado con ... 232
fret-diagram-verbose ... 234
fret-diagram-verbose, marcado con ... 234
FretBoards ... 238
frigio, modo ... 15
fuente tipográfica (de notación), tamaño ... 154
fuente tipográfica (de notación), tamaño estándar de ... 155
fuente tipográfica, cambiar ... 175
fuente tipográfica, familias de ... 176
fuente tipográfica, tamaño de ... 175
fuente, establecer la familia de ... 188
fuentes tipográficas, explicación ... 186
fundamental de un acorde ... 265

G

gaita ... 262
gaita escocesa de las tierras altas ... 262
gaita, ejemplo de ... 262
glissando ... 96, 97
Glissando ... 97, 414
globo ... 161
globo de ayuda ... 161
grace ... 434, 540
GraceMusic ... 82, 438
gráfica, notación ... 182
gráficos, descripción de los objetos ... 403
gráficos, insertar ... 180, 182
GrandStaff ... 25
GrandStaff ... 131
Gregorian square neumes ligatures ... 298
gregoriano, canto, pauta de transcripción de ... 126
GregorianTranscriptionStaff ... 127
grid-line-interface ... 164
grid-point-interface ... 164
Grid_line_span_engraver ... 162, 164
Grid_point_engraver ... 162, 164
gridInterval ... 162
GridLine ... 164
GridPoint ... 164
grob ... 398
grob-interface ... 398
grobs, sobreescritura de ... 415
grobs, visibilidad de ... 415
grosor de líneas de pauta ... 133
grupeto ... 84
grupeto circular ... 513
grupeto circular invertido ... 513
grupo especial ... 35
grupos especiales ... 33
grupos especiales, formato de ... 33
guía, formateo de las notas ... 151
guía, notas ... 151
guía, notas, formato de ... 151
guías ... 148
guiones ... 84, 198
guitarra, cabezas de nota ... 27
guitarra, tablatura de ... 223

H

Hairpin ... 90
Hal Leonard ... 28
harps ... 218
head-separation ... 346
header ... 324
hideKeySignature ... 262
hideNotes ... 158
hideStaffSwitch ... 211
horizontal, alineación de texto ... 177
horizontal, corchete ... 164
horizontal, espaciado ... 373

horizontal, papel ... 345
horizontal-bracket-interface ... 165
horizontal-shift ... 348
Horizontal_bracket_engraver ... 164, 165
HorizontalBracket ... 165
horizontally centering text ... 485
How LilyPond input files work ... 320
hufnagel ... 283, 284
huge ... 154

I

I'm hearing Voices ... 116, 253
idioma, alturas en otros ... 6
idioma, nombres de nota en otros ... 6
imágenes, insertar ... 182
importing stencils into text ... 511
impresión de caracteres especiales ... 173
impresión de los nombres de acorde ... 269
impresión, orden de ... 415
improvisación ... 30
improvisationOff ... 30
improvisationOn ... 30
includePageLayoutFile ... 434, 540
including files ... 327
indent ... 146
indent ... 348, 377
indicación de compás ... 48
Indicación de compás ... 46
indicación de pulgar ... 84
indicación del tempo ... 144
indicación metronómica ... 144
indicaciones de pedal ... 84
indicaciones de pedalero de órgano ... 84
indicaciones dinámicas nuevas ... 90
iniciar líneas de pentagrama ... 133
inlining an Encapsulated PostScript image ... 499
inscripción sobre un silencio multicompás ... 43
insertar gráficos ... 182
inserting music into text ... 503
inserting PostScript directly into text ... 500
inserting URL links into text ... 501
instrument ... 320
instrument names, centering ... 145
instrument names, changing ... 146
instrument-specific-markup-interface ... 510
InstrumentName ... 148
instrumento transpositor ... 18
instrumento, cambio de ... 147
instrumento, nombre abreviado ... 145
instrumento, nombre del ... 145
instrumento, nombres de ... 335
instrumentos transpositores ... 10
instrumentSwitch ... 434, 540
interfaz de la presentación ... 398
interna, documentación ... 403
interno, almacenamiento ... 438
intervalo ... 4
invisible, plica ... 160
invisible, silencio ... 41
invisibles, notas ... 158
Invocar a LilyPond ... 334
ionian ... 15
item-interface ... 398

J

jazz, acordes de ... 264, 272
justificado, texto ... 179
justifying lines of text ... 512
justifying text ... 490

K

keepWithTag ... 434, 540
key ... 15, 28
key signature ... 4, 296
key-cancellation-interface ... 16
key-signature-interface ... 16
Key_engraver ... 16
Key_performer ... 16
KeyCancellation ... 16
KeyChangeEvent ... 16
KeySignature ... 16
KeySignature ... 291, 292, 296, 312
killCues ... 434, 540
kirchenpausen ... 44

L

La tipografía Feta ... 502
label ... 434, 540
laissez vibrer ... 37
laissez vibrer ... 38
LaissezVibrerTie ... 38
LaissezVibrerTieColumn ... 38
large ... 154
layout file ... 351
ledger-line-spanner-interface ... 27
Ledger_line_engraver ... 27
LedgerLineSpanner ... 27
left aligning text ... 491
left-margin ... 348
length ... 212
Length and thickness of objects ... 139, 409
letra ... 191
letra asignada a una voz ... 113
letra de ensayo, estilo ... 77
letra de ensayo, formato ... 77
letra y barrado ... 58
letra y melodías ... 195
Letra, aumentar el espaciado ... 201
letra, desplazamiento de ... 41
letra, identificadores de ... 194
letras de ensayo ... 77
lidio, modo ... 15
ligadura continua ... 92
ligadura de expresión ... 93
ligadura de expresión, estilo ... 92
ligadura de fraseo ... 92
ligadura de puntos ... 92
ligadura de unión ... 36
ligadura de unión ... 38, 53
ligadura discontinua ... 92
ligadura, alteraciones y ... 5
ligaduras de expresión ... 92
ligaduras de expresión simultáneas ... 92
ligaduras de expresión y repeticiones ... 105
ligaduras de expresión, debajo de las notas ... 92
ligaduras de expresión, encima de las notas ... 92

ligaduras de expresión, posicionamiento manual ... 92
ligaduras de expresión, varias ... 92
ligaduras de fraseo ... 93
ligaduras de unión y corchetes de primera y segunda vez ... 37
ligaduras de unión, repeticiones y ... 37
ligaduras en las repeticiones ... 103
ligaduras en primera y segunda vez ... 103
ligaduras laissez vibrer ... 37
ligaduras, apariencia ... 37
ligaduras, colocación ... 37
ligaduras, en el texto ... 196
ligaduras, en la letra ... 192
Ligature_bracket_engraver ... 292, 293
LigatureBracket ... 285
Ligatures ... 285
ligatures in text ... 486
line-spanner-interface ... 414
line-width ... 348, 377
línea ... 135
línea adicional ... 135
línea de cambio de pentagrama ... 211
línea de extensión ... 198
línea de seguimiento de pentagrama ... 211
línea vertical entre pentagramas ... 162
línea, saltos ... 71
línea, saltos de ... 57, 352
líneas de rejilla ... 162
líneas divisorias ... 70
líneas divisorias invisibles ... 71
líneas divisorias, eliminación ... 419
líneas divisorias, quitar ... 49
líneas divisorias, símbolos sobre las ... 167
líneas, número de, de la pauta ... 133
LineBreakEvent ... 353
lista de colores ... 460
listener ... 542
llave ... 131
llave o corchete ... 131
llave vertical ... 127
llaves, anidado de ... 131
locrian ... 15
locrio, modo ... 15
longa ... 32
longa ... 39
lowering text ... 492
ly:add-file-name-alist ... 542
ly:add-interface ... 542
ly:add-listener ... 542
ly:add-option ... 542
ly:all-grob-interfaces ... 542
ly:all-options ... 542
ly:all-stencil-expressions ... 542
ly:assoc-get ... 542
ly:book-add-bookpart! ... 542
ly:book-add-score! ... 542
ly:book-process ... 543
ly:book-process-to-systems ... 543
ly:box? ... 543
ly:bp ... 543
ly:bracket ... 543
ly:broadcast ... 543
ly:camel-case->lisp-identifier ... 543
ly:chain-assoc-get ... 543
ly:clear-anonymous-modules ... 543
ly:cm ... 543
ly:command-line-code ... 543
ly:command-line-options ... 543
ly:command-line-verbose? ... 543
ly:connect-dispatchers ... 543
ly:context-event-source ... 543
ly:context-events-below ... 543
ly:context-find ... 543
ly:context-grob-definition ... 544
ly:context-id ... 544
ly:context-name ... 544
ly:context-now ... 544
ly:context-parent ... 544
ly:context-property ... 544
ly:context-property-where-defined ... 544
ly:context-pushpop-property ... 544
ly:context-set-property! ... 544
ly:context-unset-property ... 544
ly:context? ... 544
ly:default-scale ... 544
ly:dimension? ... 544
ly:dir? ... 544
ly:duration->string ... 544
ly:duration-dot-count ... 544
ly:duration-factor ... 544
ly:duration-length ... 544
ly:duration-log ... 544
ly:duration<? ... 545
ly:duration? ... 545
ly:effective-prefix ... 545
ly:error ... 545
ly:eval-simple-closure ... 545
ly:event-deep-copy ... 545
ly:event-property ... 545
ly:event-set-property! ... 545
ly:expand-environment ... 545
ly:export ... 545
ly:find-accidentals-simple ... 545
ly:find-file ... 545
ly:font-config-add-directory ... 545
ly:font-config-add-font ... 545
ly:font-config-display-fonts ... 545
ly:font-config-get-font-file ... 545
ly:font-design-size ... 545
ly:font-file-name ... 546
ly:font-get-glyph ... 546
ly:font-glyph-name-to-charcode ... 546
ly:font-glyph-name-to-index ... 546
ly:font-index-to-charcode ... 546
ly:font-magnification ... 546
ly:font-metric? ... 546
ly:font-name ... 546
ly:font-sub-fonts ... 546
ly:format ... 546
ly:format-output ... 546
ly:get-all-function-documentation ... 546
ly:get-all-translators ... 546
ly:get-glyph ... 547
ly:get-listened-event-classes ... 547
ly:get-option ... 547
ly:gettext ... 547
ly:grob-alist-chain ... 547
ly:grob-array-length ... 547
ly:grob-array-ref ... 547
ly:grob-array? ... 547

ly:grob-basic-properties 547
ly:grob-common-refpoint 547
ly:grob-common-refpoint-of-array 547
ly:grob-default-font 547
ly:grob-extent 547
ly:grob-interfaces 547
ly:grob-layout 547
ly:grob-object 547
ly:grob-original 547
ly:grob-parent 547
ly:grob-pq<? 548
ly:grob-properties 548
ly:grob-property 548
ly:grob-property-data 548
ly:grob-relative-coordinate 548
ly:grob-robust-relative-extent 548
ly:grob-script-priority-less 548
ly:grob-set-property! 548
ly:grob-staff-position 548
ly:grob-suicide! 548
ly:grob-system 548
ly:grob-translate-axis! 548
ly:grob? 548
ly:gulp-file 548
ly:hash-table-keys 548
ly:inch 548
ly:input-both-locations 548
ly:input-file-line-char-column 548
ly:input-location? 549
ly:input-message 549
ly:interpret-music-expression 549
ly:interpret-stencil-expression 549
ly:intlog2 549
ly:is-listened-event-class 549
ly:item-break-dir 549
ly:item? 549
ly:iterator? 549
ly:lexer-keywords 549
ly:lily-lexer? 549
ly:lily-parser? 549
ly:make-book 549
ly:make-book-part 549
ly:make-dispatcher 549
ly:make-duration 549
ly:make-global-context 550
ly:make-global-translator 550
ly:make-listener 550
ly:make-moment 550
ly:make-music 550
ly:make-music-function 550
ly:make-output-def 550
ly:make-page-label-marker 550
ly:make-page-permission-marker 550
ly:make-pango-description-string 550
ly:make-paper-outputter 550
ly:make-pitch 550
ly:make-prob 550
ly:make-scale 550
ly:make-score 551
ly:make-simple-closure 551
ly:make-stencil 551
ly:make-stream-event 551
ly:message 551
ly:minimal-breaking 356
ly:minimal-breaking 551
ly:mm 551
ly:module->alist 551
ly:module-copy 551
ly:modules-lookup 551
ly:moment-add 551
ly:moment-div 551
ly:moment-grace-denominator 551
ly:moment-grace-numerator 551
ly:moment-main-denominator 551
ly:moment-main-numerator 551
ly:moment-mod 552
ly:moment-mul 552
ly:moment-sub 552
ly:moment<? 552
ly:moment? 552
ly:music-compress 552
ly:music-deep-copy 552
ly:music-duration-compress 552
ly:music-duration-length 552
ly:music-function-extract 552
ly:music-function? 552
ly:music-length 552
ly:music-list? 552
ly:music-mutable-properties 552
ly:music-output? 552
ly:music-property 552
ly:music-set-property! 552
ly:music-transpose 552
ly:music? 552
ly:note-head::stem-attachment 553
ly:number->string 553
ly:optimal-breaking 355
ly:optimal-breaking 553
ly:option-usage 553
ly:otf->cff 553
ly:otf-font-glyph-info 553
ly:otf-font-table-data 553
ly:otf-font? 553
ly:otf-glyph-list 553
ly:output-def-clone 553
ly:output-def-lookup 553
ly:output-def-parent 553
ly:output-def-scope 553
ly:output-def-set-variable! 553
ly:output-def? 553
ly:output-description 553
ly:output-formats 553
ly:outputter-close 553
ly:outputter-dump-stencil 554
ly:outputter-dump-string 554
ly:outputter-output-scheme 554
ly:outputter-port 554
ly:page-marker? 554
ly:page-turn-breaking 355
ly:page-turn-breaking 554
ly:pango-font-physical-fonts 554
ly:pango-font? 554
ly:paper-book-pages 554
ly:paper-book-paper 554
ly:paper-book-performances 554
ly:paper-book-scopes 554
ly:paper-book-systems 554
ly:paper-book? 554
ly:paper-fonts 554
ly:paper-get-font 554

ly:paper-get-number ... 554
ly:paper-outputscale ... 554
ly:paper-score-paper-systems ... 554
ly:paper-system-minimum-distance ... 555
ly:paper-system? ... 555
ly:parse-file ... 555
ly:parser-clear-error ... 555
ly:parser-clone ... 555
ly:parser-define! ... 555
ly:parser-error ... 555
ly:parser-has-error? ... 555
ly:parser-lexer ... 555
ly:parser-lookup ... 555
ly:parser-output-name ... 555
ly:parser-parse-string ... 555
ly:parser-set-note-names ... 555
ly:performance-write ... 555
ly:pfb->pfa ... 555
ly:pitch-alteration ... 555
ly:pitch-diff ... 555
ly:pitch-negate ... 555
ly:pitch-notename ... 555
ly:pitch-octave ... 556
ly:pitch-quartertones ... 556
ly:pitch-semitones ... 556
ly:pitch-steps ... 556
ly:pitch-transpose ... 556
ly:pitch<? ... 556
ly:pitch? ... 556
ly:position-on-line? ... 556
ly:prob-immutable-properties ... 556
ly:prob-mutable-properties ... 556
ly:prob-property ... 556
ly:prob-property? ... 556
ly:prob-set-property! ... 556
ly:prob-type? ... 556
ly:prob? ... 556
ly:programming-error ... 556
ly:progress ... 556
ly:property-lookup-stats ... 556
ly:protects ... 557
ly:pt ... 557
ly:register-stencil-expression ... 557
ly:relative-group-extent ... 557
ly:reset-all-fonts ... 557
ly:round-filled-box ... 557
ly:round-filled-polygon ... 557
ly:run-translator ... 557
ly:score-add-output-def! ... 557
ly:score-embedded-format ... 557
ly:score-error? ... 557
ly:score-header ... 557
ly:score-music ... 557
ly:score-output-defs ... 557
ly:score-set-header! ... 557
ly:score? ... 557
ly:set-default-scale ... 557
ly:set-grob-creation-callback ... 558
ly:set-grob-modification-callback ... 558
ly:set-middle-C! ... 558
ly:set-option ... 558
ly:set-point-and-click ... 558
ly:set-property-cache-callback ... 558
ly:simple-closure? ... 558
ly:skyline-pair? ... 558
ly:skyline? ... 558
ly:smob-protects ... 558
ly:solve-spring-rod-problem ... 558
ly:source-file? ... 558
ly:spanner-bound ... 558
ly:spanner-broken-into ... 559
ly:spanner? ... 559
ly:staff-symbol-line-thickness ... 559
ly:start-environment ... 559
ly:stderr-redirect ... 559
ly:stencil-add ... 559
ly:stencil-aligned-to ... 559
ly:stencil-combine-at-edge ... 559
ly:stencil-empty? ... 559
ly:stencil-expr ... 559
ly:stencil-extent ... 559
ly:stencil-fonts ... 559
ly:stencil-in-color ... 559
ly:stencil-rotate ... 559
ly:stencil-rotate-absolute ... 559
ly:stencil-translate ... 559
ly:stencil-translate-axis ... 559
ly:stencil? ... 560
ly:stream-event? ... 560
ly:string-substitute ... 560
ly:system-font-load ... 560
ly:system-print ... 560
ly:system-stretch ... 560
ly:text-dimension ... 560
ly:text-interface::interpret-markup ... 560
ly:translator-description ... 560
ly:translator-group? ... 560
ly:translator-name ... 560
ly:translator? ... 560
ly:transpose-key-alist ... 560
ly:truncate-list! ... 560
ly:ttf->pfa ... 560
ly:ttf-ps-name ... 561
ly:unit ... 561
ly:usage ... 561
ly:version ... 561
ly:warning ... 561
ly:wide-char->utf-8 ... 561
lydian ... 15
LyricCombineMusic ... 194, 197
LyricExtender ... 199
LyricHyphen ... 199
Lyrics ... 142, 194, 196, 519
LyricSpace ... 193
LyricText ... 193, 207

M

m ... 265
magnifying text ... 479
magstep ... 154, 409
maj ... 265
major ... 15
major seven symbols ... 273
majorSevenSymbol ... 272
make-dynamic-script ... 91
make-pango-font-tree ... 188
makeClusters ... 112, 434, 540
manetener música etiquetada ... 330

mantenimiento, pedal, estilos de ... 214
manual, cambio de pentagrama ... 209
manual, marca de repetición ... 105
manuales, barras ... 56
maqam ... 309
maqams ... 309
marca de ensayo, estilo ... 77
marca de ensayo, formato ... 77
marcado ... 173
marcado de texto ... 173
marcado, caracteres especiales en modo de ... 173
marcado, definir instrucciones de ... 443
marcado, expresiones de ... 173
marcado, sintaxis ... 173
marcas de fraseo ... 93
marcato ... 85, 513
matices ... 86
matices absolutos ... 86
matices, posición vertical ... 87
matices, varios sobre una nota ... 87
maxima ... 39
mayor, modo ... 15
`measureLength` ... 83
`measurePosition` ... 48
`measurePosition` ... 83
Medicaea, Editio ... 283, 284
medida, música sin ... 83
medios, intervalos ... 309
melisma ... 197, 198
melismata ... 197
menor, modo ... 15
mensuración, símbolo de ... 288
mensural ... 283, 284
Mensural ligatures ... 292
mensural, música, transcripción de ... 130
`Mensural_ligature_engraver` ... 292, 293, 294
`MensuralStaff` ... 127
MensuralStaffContext ... 286
MensuralVoiceContext ... 286
mensurstriche, diseño ... 130
`mergeDifferentlyDottedOff` ... 116
`mergeDifferentlyDottedOn` ... 116
`mergeDifferentlyHeadedOff` ... 116
`mergeDifferentlyHeadedOn` ... 116
merging text ... 486
`meter` ... 320
meter, polymetric ... 50
`metro` ... 52
metro, medida ... 46
`MetronomeMark` ... 144
`metrónomo` ... 144
metrónomo, indicación de ... 142
metrónomo, indicación de, con texto ... 142
mezclar notas ... 116
mezzosoprano, clave de ... 12
`mf` ... 86
microtonos ... 7
microtonos en MIDI ... 338
MIDI ... 17, 335
MIDI, alturas ... 338
MIDI, cuartos de tono ... 338
MIDI, definiciones de contexto ... 337
MIDI, duraciones ... 338
MIDI, microtonos ... 338
MIDI, nombres de ... 338
MIDI, transposición ... 17
`minimumFret` ... 225
`minimumPageTurnLength` ... 356
`minimumRepeatLengthForPageTurn` ... 356
`minor` ... 15
`mixed` ... 214
mixolidio, modo ... 15
`mixolydian` ... 15
`modern` ... 21
`modern-cautionary` ... 21
`modern-voice` ... 21
`modern-voice-cautionary` ... 22
moderno de precaución, alteraciones de estilo ... 21
moderno, alteraciones de estilo ... 21
moderno, estilo de alteraciones ... 20
moderno-precaución, estilo de alteraciones ... 20
modificadores de acorde ... 265
modo de acordes ... 264
`modo eclesiástico` ... 16
modos ... 15
modos eclesiásticos ... 15
mordent ... 513
mordente ... 84
mordente circular ... 84
mordente circular, inverso ... 84
mordentes ... 78, 262
movimientos, varios ... 317
`Moving objects` ... 177, 180, 209
`mp` ... 86
multicompás, posicionamiento de los silencios ... 44
multicompás, silencio, adjuntar texto ... 43
multicompás, silencio, aplicar un calderón a ... 43
multicompás, silencios ... 42
multicompás, silencios, contracción de ... 42
multicompás, silencios, expansión de ... 42
multicompás, silencios, inscripción en ... 43
`MultiMeasureRest` ... 45
`MultiMeasureRestNumber` ... 45
`MultiMeasureRestText` ... 45
`Multiple notes at once` ... 119
multiple phrasing slurs ... 94
`Music classes` ... 437
`Music expressions` ... 437
`Music expressions explained` ... 317
`Music properties` ... 438
música desplegada con finales alternativos ... 107
musica ficta, alteraciones ... 292
música intercalada ... 123
música para principiantes ... 28
música sin medida ... 83
música sin metro ... 49
`musicMap` ... 434, 540
musicológico, análisis ... 164

N

natural, nota ... 4
naturales, armónicos ... 221
`neo-modern` ... 23
`neo-modern-cautionary` ... 23
neo-moderno de precaución, estilo de alteraciones ... 23
neo-moderno, estilo de alteraciones ... 23
neomensural ... 284

nested contexts ... 396
Nesting music expressions ... 135, 139
New_fingering_engraver ... 157, 397
niente, al ... 88
no restablecer, estilo de alteraciones ... 24
no vacíos, textos ... 166
no-reset ... 24
Nombre de las duraciones de notas y silencios ... 32
nombre del cantante ... 203
nombres de las notas ... 1
Nombres de las notas ... 2, 4, 6, 7
noPageBreak ... 434, 540
noPageTurn ... 434, 540
normalsize ... 154
nota abierta ... 84
nota de espaciado ... 41
nota tapada ... 84
nota, cabeza de, armónico ... 27
nota, cabeza de, en aspas ... 27
nota, cabeza de, en parlato ... 27
nota, cabeza de, en rombo ... 27
nota, cabeza de, estilos ... 27
nota, cabeza de, guitarra ... 27
nota, cabeza especial de ... 27
nota, estilos de cabeza ... 475
notación, explicación de ... 161
notación, tamaño de la fuente ... 154
notas coloreadas ... 158
notas coloreadas en acordes ... 159
notas de adorno ... 81
notas de adorno ... 262
notas de pentagrama cruzado ... 212
notas entre paréntesis ... 160
notas fantasma ... 160
notas guía ... 148, 151
notas invisibles ... 158
notas ocultas ... 158
notas transparentes ... 158
notas, cabezas de ... 154
notas, cabezas de, Aiken ... 28
notas, cabezas de, arpa sacra ... 28
notas, cabezas de, barradas ... 30
notas, cabezas de, con formas ... 28
notas, cabezas de, estudio ... 28
notas, cabezas de, improvisación ... 30
notas, cabezas de, notación simplificada ... 28
notas, colisiones de ... 116
notas, con puntillo ... 32
notas, corchetes de agrupación de ... 164
notas, división de ... 53
notas, duración de ... 31
notas, nombres de, en otros idiomas ... 6
notas, nombres holandeses de ... 4
notas, nombres predeterminados ... 4
notas, transporte de ... 9
note-collision-interface ... 530, 532, 533
note-event ... 27, 28, 30
note-event ... 150
note-head-interface ... 27, 28, 30
Note_head_line_engraver ... 212
Note_heads_engraver ... 27, 28, 30, 53, 394
Note_spacing_engraver ... 158
NoteCollision ... 119
NoteColumn ... 119
NoteEvent ... 437, 438
NoteHead ... 27, 28, 30, 289, 448
notes de pentagrama cruzado ... 209
notes within text by log and dot-count ... 503
notes within text by string ... 503
NoteSpacing ... 158, 374
nots, cabezas de, para precticar ... 28
nuevo pentagrama ... 126
nuevos, contextos ... 390
numeración de compases, quitar ... 49
número de compás ... 83
número de compases, formato de ... 74
número de líneas de pauta ... 133
números de compás ... 73
números de compás y repeticiones ... 105
números de cuerda frente a digitaciones ... 224

O

objects, rotating ... 420
objetos, coloreado de ... 415
objetos, colorear ... 158
objetos, sobreescritura de ... 415
objetos, visibilidad de ... 415
ocatva, especificación relativa ... 2
octava, comprobación de ... 8
octava, corrección de ... 8
octava, especificación absoluta ... 1
octava, introducción absoluta ... 1
octava, introducción relativa ... 2
octava, marca de cambio de ... 1
octava, transposición de ... 12
octavación ... 16
octavación ... 17
octavadas, visibilidad de las claves ... 419
OctavateEight ... 14
octaveCheck ... 8
octaveCheck ... 434, 540
ocultación de pentagramas ... 139
ocultar pentagramas ... 139
ocultar pentagramas de música antigua ... 140
ocultar pentagramas rítmicos ... 140
ocultas, notas ... 158
oddFooterMarkup ... 324
oddHeaderMarkup ... 323
oldaddlyrics ... 434, 540
olvidar, estilo de alteraciones ... 24
On the un-nestedness of brackets and ties ... 93, 94
oneVoice ... 113
open ... 513
opus ... 320
organ pedal marks ... 513
Organizing pieces with variables ... 125, 328, 332
órgano, indicaciones de pedal de ... 513
orientación ... 345
ornamentación ... 84
ornamentos ... 78, 84
orquestales, cuerdas ... 220
ossia ... 135
ossia ... 139
ossia ... 140, 395
Other sources of information ... 59, 62, 116, 327, 328, 337, 339, 421

`Other uses for tweaks` ... 208, 209
`ottava` ... 16, 434, 540
`ottava-bracket-interface` ... 17
`Ottava_spanner_engraver` ... 17
`OttavaBracket` ... 17
`outside-staff-horizontal-padding` ... 372
`outside-staff-padding` ... 372
`outside-staff-priority` ... 372
`overrideProperty` ... 434, 540
`OverrideProperty` ... 401
overriding properties within text markup ... 510

P

`p` ... 86
padding text ... 493
padding text horizontally ... 493
`page-breaking-between-system-padding` ... 349
`page-count` ... 349
`page-limit-inter-system-space` ... 349
`page-limit-inter-system-space-factor` ... 349
`page-spacing-weight` ... 349
`page-top-space` ... 346
`pageBreak` ... 434, 540
`pageTurn` ... 434, 540
página, disposición de la ... 377
página, saltos de ... 377
página, saltos, forzar ... 320
página, tamaño de la ... 345
Pango ... 186
`pantagrama` ... 127, 135, 139
papel, tamaño del ... 345
`paper-height` ... 347
`paper-width` ... 348
paralelo, música en ... 123
`parallelMusic` ... 123
`parallelMusic` ... 435, 541
parcial, compás ... 48
paréntesis ... 160
paréntesis en ángulo ... 111
paréntesis, alteración con ... 5
`parentheses-interface` ... 160
`ParenthesesItem` ... 160
`Parenthesis_engraver` ... 160
`parenthesize` ... 160
`parenthesize` ... 435, 541
parlato ... 191
parlato, cabezas de nota ... 27
`partcombine` ... 120
`partcombine` ... 435, 541
`PartCombineMusic` ... 123
`parte` ... 123
parte al alzar ... 48
partes, combinación automática ... 120
partes, combinador de ... 120
pausa, marcas de ... 94
pauta ... 126
pauta de coro ... 127
pauta de percusión ... 126
pauta, escalado de ... 135
pedal de mantenimiento, estilo de ... 214
pedal de órgano ... 84
pedal derecho ... 214
pedal forte ... 214
pedal sostenuto ... 214
pedal, estilos de indicación ... 214
pedal, indicación mixta ... 214
pedal, indicación por corchete ... 214
pedal, indicación textual ... 214
pedal, indicaciones de ... 513
pedales de piano ... 214
pedales del arpa ... 219
pedales del arpa, diagramas ... 219
`pedalSustainStyle` ... 214
pentagrama a la francesa ... 135
pentagrama cruzado, arpegios con paréntesis ... 100
pentagrama cruzado, plica de ... 212
pentagrama cruzado, trémolo de ... 111
pentagrama único ... 126
pentagrama único, polifonía de ... 113
pentagrama vacío ... 139
pentagrama, ajuste del símbolo del ... 133
pentagrama, cambio ... 211
pentagrama, cambio automático de ... 210
pentagrama, cambio de ... 211
pentagrama, cambios manuales de ... 209
pentagrama, distancia entre los ... 360
pentagrama, establecer el símbolo del ... 409
pentagrama, establecer tamaño de ... 351
pentagrama, grosor de las líneas ... 133
pentagrama, iniciar ... 126
pentagrama, instanciar ... 126
pentagrama, línea de cambio de ... 211
pentagrama, nuevo ... 126
pentagrama, número de líneas ... 133
pentagrama, ocultación de ... 139
`pentagramas` ... 127
pentagramas anidados ... 131
pentagramas de instrumentos de teclado ... 208
pentagramas de piano ... 127, 208
pentagramas, grupo de ... 127
pentagramas, varios ... 127
`percent` ... 108
`PercentRepeat` ... 109
`PercentRepeatCounter` ... 109
`PercentRepeatedMusic` ... 109
percusión ... 251, 253
percusión, pauta de ... 126
personalización de los nombres de acorde ... 272
personalizados, añadir diagramas de traste ... 240
personalizados, diagramas de trastes ... 229, 236
Petrucci ... 283, 284
phrasing slurs, multiple ... 94
phrasing slurs, simultaneous ... 94
`PhrasingSlur` ... 94
`phrasingSlurDashed` ... 93
`phrasingSlurDotted` ... 93
`phrasingSlurDown` ... 93
`phrasingSlurNeutral` ... 93
`phrasingSlurSolid` ... 93
`phrasingSlurUp` ... 93
`phrygian` ... 15
`piano` ... 22
piano de precaución, alteraciones ... 22
piano de precaución, estilo de alteraciones ... 22
`Piano templates` ... 209
piano, alteraciones de ... 22
piano, dinámica centrada para música de ... 209
piano, estilo de alteraciones de ... 22

piano, pedales de ... 214
piano, pentagramas de ... 127, 208
piano, sistema de ... 127
piano-cautionary ... 22
Piano_pedal_engraver ... 215
PianoPedalBracket ... 215
PianoStaff ... 25, 100
PianoStaff ... 131
PianoStaff ... 148, 208, 209, 210
picado ... 85
pie ... 324
piece ... 320
pipeSymbol ... 76
Pitch_squash_engraver ... 30, 56, 394, 523
pitched trills ... 100
pitchedTrill ... 100
pitchedTrill ... 435, 541
pizzicato, Bartók ... 222
pizzicato, snap ... 222
placing horizontal brackets around text ... 500
placing vertical brackets around text ... 498
plantilla música árabe ... 314
plica ... 160
plica de pentagrama cruzado ... 212
plica invisible ... 160
plicas de pentagrama cruzado ... 212
poet ... 320
pointAndClickOff ... 435, 541
pointAndClickOn ... 435, 541
polifonía ... 119
polifonía en un solo pentagrama ... 113
polifónica, música ... 116
polimétrica ... 35, 52
polimétricas, partituras ... 392
polymetric signatures ... 50
porcentaje, repetición de ... 108
portato ... 85
portato ... 86
portato ... 513
posición y cejilla, indicación para instrumentos de trastes ... 250
PostScript ... 182
pp ... 86
ppp ... 86
pppp ... 86
prall ... 513
prallmordent ... 84, 513
prallprall ... 84, 513
precaución, alteración de ... 5
precaución, alteraciones de piano de ... 22
precaución, estilo de alteraciones, piano ... 22
predefinedFretboardsOff ... 246
predefinedFretboardsOn ... 246
predefinidas, afinaciones, para instrumentos de traste ... 228
predeterminado, estilo de las alteraciones ... 19
predeterminados, nombres de nota ... 4
presentación, interfaz de la ... 398
prima volta ... 102
primera y segunda vez ... 102
primera y segunda vez ... 105
primera y segunda vez con ligaduras ... 103
primera y segunda vez, corchetes y ligaduras en ... 37
principiantes, música para ... 28
print-all-headers ... 323, 349
print-first-page-number ... 349
print-page-number ... 349
PropertySet ... 401
propiedades ... 401
pulsos por minuto ... 142
pulsos, agrupar ... 61
puntillo, notas con ... 32
puntos, ligadura de ... 92
puntuación ... 191
putting space around text ... 493

Q

quebrado, acorde ... 97
quinta ... 4
quitar música etiquetada ... 330
quotedEventTypes ... 150
quoteDuring ... 435, 541
QuoteMusic ... 151

R

r ... 39
R ... 42
racimo (cluster) ... 112
racimo (cluster) ... 113
ragged-bottom ... 349
ragged-last ... 349, 377
ragged-last-bottom ... 349
ragged-right ... 350, 377
raising text ... 494
rango de alturas ... 25
Ratisbona, Editio ... 284
Real music example ... 208, 209
recordatoria, alteración ... 5
redonda, silencios de, para un compás completo ... 42
Referencia de funcionamiento interno ... 388
referencing page numbers in text ... 510
regulador ... 86
regulador ... 90
reguladores en ángulo ... 420
regular line breaks ... 353
RehearsalMark ... 78, 171
rejilla, líneas de ... 162
relativa, altura en acordes ... 111
relativa, especificación de la octava ... 2
relativa, introducción de la octava, y acordes ... 3
relativas, especificación de octavas ... 2
relative ... 2, 4, 11, 210
RelativeOctaveCheck ... 9
RelativeOctaveMusic ... 4
relativo ... 2
relativo, modo, autocambiador y ... 210
relativo, modo, transposición y ... 4
relleno ... 399
relleno alrededor del texto ... 181
removeWithTag ... 435, 541
renacimiento, música del ... 130
repeatCommands ... 105
RepeatedMusic ... 105, 107, 108, 438
repeats ... 71
RepeatSlash ... 109
repetición ... 105
repetición ambigua ... 105

repetición anidada 105
repetición con anacrusa 103
repetición con finales alternativos 102
repetición corta 108
repetición de compás 108
repetición de estilo porcentaje 109
repetición de porcentaje 108
repetición de trémolo 110
repetición delplegada 107
repetición deplegada con finales alternativos 107
repetición manual 105
repetición y ligadura de expresión 105
repetición y números de compás 105
repetición, barras de 70
repetición, cambiar finales alternativos 105
repetición, cambiar número de 105
repetición, comienzo de 105
repetición, contador de tiempo de la 105
repetición, final de 105
repetición, ligaduras de unión en 37
repeticiones con ligaduras 103
repeticiones explícitas 107
repeticiones normales 102
repeticiones, expandir 338
repetitiva, música 107
reservados, impresión de caracteres 173
resetRelativeOctave 435, 541
respiraciones 94
Rest 40, 291
rest-event 150
RestCollision 119
rests, ancient 291
reverseturn 513
RevertProperty 401
rfz 86
rgb, color 159
rgb-color 159
RhythmicStaff 30, 56, 127
right aligning text 494
rightHandFinger 248
rightHandFinger 435, 541
rítmica, pauta 126
rombo, cabezas de nota 27
rotating objects 420
rotating text 495

S

s 41
sacredHarpHeads 28
salto de línea 57
saltos de línea 352
saltos de página 377
SATB 198
scaleDurations 435, 541
scaling durations 35
scaling text 495
Scheme signature 446
Scheme tutorial 388, 429
scordatura 16
Score 84, 516, 518
Score is a (single) compound musical expression 317
Scores and parts 328
scoreTitleMarkup 323
scoreTweak 435, 541
Script 86
seconda volta 102
segno 78, 84, 513
segno sobre la línea divisoria 167
seguimiento de voz 211
selección del tamaño de la fuente tipográfica (de notación) 154
self-alignment-interface 398, 421
sello, eliminar 415
Semai, forma 312
semi-bemol 7
semi-bemoles 4
Semi-flat symbol appearance 310
semi-sostenido 7
semi-sostenidos 4
semitono 6
semitrino 84, 513
semitrino ascendente 84
semitrino descendente 84, 513
separate text 171
SeparationItem 374
séptima, acordes de 265
SequentialMusic 438
sesqui-bemol 7
sesqui-sostenido 7
set-accidental-style 19
set-octavation 16
setting extent of text objects 512
setting horizontal text alignment 488
Setting simple songs 190
setting subscript in standard font size 479
setting superscript in standard font size 479
sf 86
sff 86
sfz 86
shiftDurations 435, 541
shiftOff 116
shiftOn 116
shiftOnn 116
shiftOnnn 116
short-indent 146
short-indent 348
show-available-fonts 188
showFirstLength 334
showKeySignature 262
showLastLength 334
showStaffSwitch 211
side-position-interface 398, 421
signature, Scheme 446
signatures, polymetric 50
silencio 39
silencio invisible 41
silencio normal, condensar 45
silencio, colisiones de 45
silencio, especificar la posición vertical 39
silencios de compás completo 39
Silencios de compás completo 42
silencios eclesiásticos 44
silencios multicompás 39
silencios multicompás 45
Silencios multicompás 42
silencios, indocar duraciones 39
símbolos no musicales 182

simile ... 109
simple text strings ... 481
simple text strings with tie characters ... 505
simplificada, notación ... 28
simultáneas, ligaduras de expresión ... 92
simultáneas, notas y alteraciones ... 25
simultaneous phrasing slurs ... 94
SimultaneousMusic ... 437
sistema de coro ... 127
sistema de piano ... 127
sistema de piano ... 131
sistema, delimitador de comienzo ... 127
sistema, delimitadores de inicio anidados ... 131
Sistemas a la francesa ... 139, 142
Size of objects ... 139
SkipMusic ... 41
skipTypesetting ... 334
slashed digits ... 510
slides en notación de tablatura ... 226
Slur ... 93
slurDashed ... 92
slurDotted ... 92
slurDown ... 92
slurNeutral ... 92
slurs, multiple phrasing ... 94
slurs, simultaneous phrasing ... 94
slurSolid ... 92
slurUp ... 93
small ... 154
snap pizzicato ... 222
sobreescritura de objetos ... 415
Sol, clave de ... 12
Solesmes ... 284
solo, partes ... 120
sonido ... 335
soprano, clave de ... 12
sos ... 214
sostenido ... 4
sostenido ... 6
sostenido, doble ... 4
sostenuto, pedal ... 214
SostenutoEvent ... 215
sostenutoOff ... 214
sostenutoOn ... 214
SostenutoPedal ... 214
SostenutoPedalLineSpanner ... 214
sp ... 86
spacing ... 374
spacing-spanner-interface ... 536, 537
SpacingSpanner ... 373, 374, 375
spacingTweaks ... 435, 541
SpanBar ... 72
spp ... 86
Sprechgesang ... 191
Square neumes ligatures ... 298
staccatissimo ... 85, 513
staccato ... 85
staccato ... 86
staccato ... 513
stacking text in a column ... 485
Staff ... 25, 27, 52, 72, 127
Staff ... 131
Staff ... 142, 148, 165, 286, 374, 516
staff-padding ... 209
staff-symbol-interface ... 134, 135
Staff.midiInstrument ... 335
Staff_symbol_engraver ... 142
StaffGroup ... 75
StaffGroup ... 131
StaffGroup ... 132
StaffSpacing ... 374
StaffSymbol ... 127, 135, 139, 351
standalone text ... 171
StanzaNumber ... 207
start-repeat ... 105
startGroup ... 164
startTrillSpan ... 100
Stem ... 161, 212, 214, 290, 448
stem, direction ... 160
stem, down ... 160
stem, neutral ... 160
stem, up ... 160
stem, with slash ... 80
stem-interface ... 161
stem-spacing-correction ... 374
Stem_engraver ... 161
stemDown ... 160
stemLeftBeamCount ... 67
stemNeutral ... 160
stemRightBeamCount ... 67
stemUp ... 160
stopGroup ... 164
stopped ... 513
stopTrillSpan ... 100
storePredefinedDiagram ... 241
storePredefinedDiagram ... 435, 541
String quartet ... 220
StringNumber ... 225
stringTunings ... 238
StringTunings ... 228
StrokeFinger ... 249
styles, voice ... 116
subdivideBeams ... 61
subíndice ... 175
subscript text ... 482
subsubtitle ... 320
subtitle ... 320
suggestAccidentals ... 292
superíndice ... 175
superscript text ... 483
sus ... 268
SustainEvent ... 214
sustainOff ... 214
sustainOn ... 214
SustainPedal ... 214
SustainPedalLineSpanner ... 214
system ... 127
system-count ... 350
system-separator-markup ... 350
SystemStartBar ... 131
SystemStartBar ... 132
SystemStartBrace ... 131
SystemStartBrace ... 132
SystemStartBracket ... 131
SystemStartBracket ... 132
SystemStartSquare ... 131
SystemStartSquare ... 132

T

Tab_note_heads_engraver ... 229
tablatura ... 223
tablatura de bajo ... 228
tablatura de banjo ... 223, 228, 250
tablatura de mandolina ... 228
tablatura predeterminada ... 225
tablatura y armónicos ... 226
tablatura y slides ... 226
tablatura, afinaciones predefinidas de ... 228
tablatura, pauta de ... 126
tablatura, principios básicos ... 225
tablaturas personalizadas ... 228
TabNoteHead ... 227
TabStaff ... 127, 225, 227
TabVoice ... 225, 227
tag ... 435, 541
tagline ... 321
tagline ... 323
taor ... 262
tapadas, notas, en instrumentos de trastes ... 250
taqasim ... 312
teaching ... 23
teaching (enzeñanza), estilo de alteraciones ... 23
tecla, pentagramas para instrumentos de ... 208
teclado, matices centrados para música de ... 209
teclado, pentagramas para instrumentos de ... 208
teeny ... 154
tempo ... 142
tenor coral, clave ... 12
tenor, clave de ... 12
tenuto ... 85
tenuto ... 86
tenuto ... 513
tesitura ... 25
tesitura ... 26
text ... 214
text columns, left-aligned ... 492
text columns, right-aligned ... 494
text spanners ... 167
text, standalone ... 171
text-interface ... 398, 510
text-script-interface ... 398
texto alineación de ... 177
texto de varias líneas ... 179
texto destacado ... 166
texto en casilla de repetición ... 106
texto entrecomillado en modo de marcado ... 173
texto sobre un silencio multicompás ... 43
texto, alineación horizontal ... 177
texto, alineación vertical ... 177
texto, elementos no vacíos de ... 166
texto, en otros idiomas ... 166
Texto, inscripciones de ... 166
texto, marcado de ... 173
texto, relleno de ... 181
texto, tamaño de ... 175
TextScript ... 86, 166, 173, 177, 180, 183, 185, 186
TextSpanner ... 167, 414
textSpannerDown ... 167
textSpannerNeutral ... 167
textSpannerUp ... 167
Thorough bass ... 276
thumb ... 156
thumb marking ... 513
thumb-script ... 156
Tie ... 38, 450
TieColumn ... 38
tiempo, administración ... 83
tiempo, control del (dentro de la partitura) ... 83
tiempo, repeticiones y contador de ... 105
ties and chords ... 37
TimeScaledMusic ... 35
TimeSignature ... 48
TimeSignature ... 52, 289
Timing_translator ... 48
Timing_translator ... 49, 52, 72, 84, 518
tiny ... 154
tipografía Feta ... 461
tipografía, establecer tamaño de ... 351
tipografiar texto ... 173
title ... 320
títulos ... 324
tocItem ... 436, 542
Top ... 388
top-level text ... 171
top-margin ... 347
transcripción de música mensural ... 130
translating text ... 495
Translation ... 397
transparentes, hacer los objetos ... 415
transparentes, notas ... 158
transporte ... 9
transporte de diagramas de trastes ... 239
transpose ... 4, 9, 11
transposedCueDuring ... 436, 542
TransposedMusic ... 11
transposición ... 9
transposición de alturas ... 9
transposición de MIDI ... 17
transposición de notas ... 9
transposición instrumental ... 17
transposición y modo relativo ... 4
transposition ... 17
transposition ... 436, 542
transpositor, instrumento ... 17
transpositoras, claves ... 12
transpositores, instrumentos ... 10
traste ... 225
trastes personalizados, añadir diagramas de ... 240
trastes, añadir digitaciones a diagramas de ... 247
trastes, armónicos en instrumentos de ... 250
trastes, diagramas automáticos de ... 245
trastes, diagramas de ... 229
trastes, diagramas de, con nombres de acorde ... 238
trastes, diagramas de, personalizados ... 229
trastes, diagramas de, transporte ... 239
trastes, diagramas personalizados de ... 236
trastes, instrumentos con, afinaciones predefinidas ... 228
trastes, instrumentos de, digitaciones de la derecha ... 248
trastes, instrumentos de, formas de acordes ... 241
trastes, instrumentos de, indicacion de la posición y cejilla ... 250
trastes, notas tapadas en instrumentos de ... 250
tre corde ... 214
treCorde ... 214
tremolo ... 110

trémolo, indicaciones de ... 110
tremoloFlags ... 110
trémols de pentagrama cruzado ... 111
tresillo ... 35
tresillos ... 33
tresillos, formato de ... 33
tríadas ... 265
trill ... 100
trill ... 513
trills, pitched ... 100
TrillSpanner ... 101, 414
trino ... 84
trino ... 101
trino ... 513
trino con nota y alteración forzada ... 101
trinos ... 100
trucar ... 403
trucar puntos de control ... 405
trucos ... 403
trucos en la letra ... 405
trucos en una variable ... 405
TupletBracket ... 35
TupletNumber ... 34
TupletNumber ... 35
tupletNumberFormatFunction ... 33
tupletSpannerDuration ... 33
turn ... 513
tweak ... 436, 542
Tweaking methods ... 35, 404, 405
Tweaking output ... 388, 421

U

U.C. ... 214
una corda ... 214
unaCorda ... 214
UnaCordaEvent ... 215
UnaCordaPedal ... 215
UnaCordaPedalLineSpanner ... 215
underlining text ... 484
unfold ... 107
UnfoldedRepeatedMusic ... 105, 108
unfoldRepeats ... 436, 542
unHideNotes ... 158
upbow ... 513

V

vacío, pentagrama ... 139
varcoda ... 84, 513
variables ... 319
variables, uso de las ... 328
varias ligaduras de expresión ... 92
varias líneas, marcado de ... 179
varias voces ... 116
varios matices sobre una nota ... 87
Vaticana, Editio ... 283, 284
Vaticana_ligature_engraver ... 294
VaticanaStaff ... 127
VaticanaStaffContext ... 294
VaticanaVoiceContext ... 294
verso, número de ... 202
vertical, alineación de texto ... 177
vertical, espaciado ... 360, 377
vertical, línea, entre pentagramas ... 162
vertical, posición, de las indicaciones dinámicas ... 87
VerticalAlignment ... 360, 362
VerticalAxisGroup ... 142, 360, 361
vertically centering text ... 496
viento, instrumentos ... 260
violín, clave de ... 12
visibilidad de las claves octavadas ... 419
visibilidad de los objetos ... 415
Visibility and color of objects ... 392, 415, 416, 419
Vocal ensembles ... 198
Vocal ensembles ... 395
voces, alteraciones en varias ... 21
voces, varias ... 116
voice ... 19, 20
Voice ... 27, 30, 113, 121, 123, 151, 154, 195, 196, 285, 292, 374, 400
voice styles ... 116
VoiceFollower ... 212, 414
voiceOne ... 113
Voices contain music ... 115, 119
volta ... 102
volta, prima ... 102
volta, seconda ... 102
Volta_engraver ... 271
VoltaBracket ... 105, 107
VoltaRepeatedMusic ... 105, 107
voz ... 113
voz, estilo de alteraciones ... 20
voz, seguimiento de ... 211

W

whichBar ... 72
White mensural ligatures ... 292
with-color ... 158
Within-staff objects ... 408
withMusicProperty ... 436, 542
Working on input files ... 317

X

x11, color de ... 158, 159
x11-color ... 158, 159