LilyPond

El tipografiador de música

Utilización del programa

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 Instalación ... 1

1.1 Archivos binarios pre-compilados ... 1

Descarga ... 1

1.2 Compilar a partir del código fuente ... 1

2 Configuración ... 2

2.1 Ajustes para sistemas operativos específicos ... 2

2.1.1 Configuración para MacOS X ... 2

2.2 Apoyo respecto de los editores de texto ... 3

2.2.1 Modo de Emacs ... 3

2.2.2 Modo de Vim ... 3

2.2.3 jEdit ... 3

2.2.4 TexShop ... 3

2.2.5 TextMate ... 3

2.2.6 LilyKDE ... 4

2.3 Apuntar y pulsar ... 4

3 Ejecutar LilyPond ... 6

3.1 Utilización normal ... 6

3.2 Utilización desde la línea de órdenes ... 6

3.2.1 Invocar `lilypond` ... 6

3.2.2 Opciones de la línea de órdenes para `lilypond` ... 6

3.2.3 Variables de entorno ... 10

3.3 Mensajes de error ... 10

3.4 Actualizar ficheros con `convert-ly` ... 11

3.4.1 Opciones de la línea de órdenes para `convert-ly` ... 11

3.4.2 Problemas con `convert-ly` ... 12

3.5 Informar de fallos ... 13

4 `lilypond-book`: integrar texto y música ... 14

4.1 Un ejemplo de documento musicológico ... 14

4.2 Integrar música y texto ... 17

4.2.1 LaTeX ... 17

4.2.2 Texinfo ... 18

4.2.3 HTML ... 19

4.2.4 DocBook ... 20

4.3 Opciones de fragmentos de música ... 21

4.4 Invocar `lilypond-book` ... 23

4.5 Extensiones de nombres de archivo ... 26

4.6 Otros métodos para mezclar texto y música ... 26

Muchas citas de una partitura extensa ... 26

Insertar la salida de LilyPond dentro de OpenOffice.org ... 26

Insertar la salida de LilyPond dentro de otros programas ... 26

5 Conversión desde otros formatos 28
5.1 Invocar `midi2ly` 28
5.2 Invocar `musicxml2ly` 29
5.3 Invocar `abc2ly` 30
5.4 Invocar `etf2ly` 31
5.5 Generar archivos de LilyPond 31

Apéndice A Licencia de documentación libre de GNU 32

Apéndice B Índice de LilyPond 38

1 Instalación

Existen dos conjuntos de ediciones o lanzamientos de LilyPond: las ediciones estables y las ediciones inestables de desarrollo. Las versiones estables tienen como número 'menor' de versión un número par (es decir: 2.8, 2.10, 2.12, etc). Las versiones de desarrollo tienen un número 'menor' de versión impar (esto es: 2.7, 2.9, 2.11, etc).

La compilación de LilyPond es un proceso muy enrevesado, por tanto recomendamos **fervientemente** el uso de los archivos binarios precompilados.

1.1 Archivos binarios pre-compilados

Descarga

Revise el enlace http://lilypond.org/web/install/ para ver información actualizada sobre los paquetes binarios para su plataforma. Si su sistema operativo no está contemplado en dicha página general, consulte la lista completa que está en http://download.linuxaudio.org/lilypond/binaries/

Por el momento estamos creando archivos binarios para

```
darwin-ppc  - MacOS X powerpc
darwin-x86  - MacOS X intel
freebsd-64  - FreeBSD 6.x, x86_64
freebsd-x86 - FreeBSD 4.x, x86
linux-64    - Cualquier distribución de GNU/Linux, x86_64
linux-ppc   - Cualquier distribución de GNU/Linux, powerpc
linux-x86   - Cualquier distribución de GNU/Linux, x86
mingw       - Windows x86
```

Advertencias y problemas conocidos

Si tiene MacOS 10.3 o 10.4 y quiere usar guiones de Python como `convert-ly` y `lilypond-book`, consulte Sección "Setup for MacOS X" in *Application Usage*.

1.2 Compilar a partir del código fuente

Es posible compilar LilyPond directamente del código fuente. Ello requiere poder leer en inglés, motivo por el que esta sección se deja sin traducir. Si verdaderamente quiere compilar LilyPond, consulte Sección "Compiling from source" in *Application Usage*.

2 Configuración

Este capítulo trata sobre varias opciones de configuración, posteriores a la instalación, para LilyPond y otros programas. Este capítulo se puede tratar tranquilamente como una referencia: lea solamente las secciones que le interesen.

2.1 Ajustes para sistemas operativos específicos

Esta sección explica cómo realizar puestas a punto adicionales para sistemas operativos específicos.

2.1.1 Configuración para MacOS X

Uso de guiones Python en MacOS 10.3 o 10.4

Los archivos binarios de LilyPond para MacOS X no incluyen Python, pero se necesita Python 2.4 o posterior por parte de `convert-ly`. Por ello, si usamos MacOS 10.3 o 10.4, debemos instalar una versión de Python más reciente desde http://python.org/download/, y luego editar la primera línea de `convert-ly` y `lilypond-book` como sigue: si el binario de Python que hemos instalado está en nuestro *PATH*, la primera línea debe ser

```
#!/usr/bin/env python
```

en caso contrario, debe ser

```
#!/path/to/newly_installed/python
```

MacOS X sobre línea de órdenes

Los «scripts» (guiones como `lilypond-book`, `convert-ly`, `abc2ly` e incluso el propio `lilypond`) van incluidos dentro del archivo de aplicación .app de MacOS X. Se pueden ejecutar a partir de la línea de órdenes invocándolos directamente, p.ej.

```
ruta/hacia/LilyPond.app/Contents/Resources/bin/lilypond
```

Lo mismo vale para otros scripts de ese directorio, incluidos los guiones `lilypond-book`, `convert-ly`, `abc2ly`, etc.

Como alternativa, puede crear scripts que añadan la ruta automáticamente. Cree un directorio para guardar estos guiones,

```
mkdir -p ~/bin
cd ~/bin
```

Cree un archivo de nombre `lilypond` que contenga

```
exec ruta/hacia/LilyPond.app/Contents/Resources/bin/lilypond "$@"
```

Cree archivos similares `lilypond-book`, `convert-ly` y cualesquiera otros programas de apoyo que vaya a utilizar (`abc2ly`, `midi2ly`, etc). Sencillamente, sustituya el fragmento `bin/lilypond` por `bin/convert-ly` (u otro nombre de programa) en el archivo anterior.

Haga ejecutable al archivo,

```
chmod u+x lilypond
```

Y ahora añada este directorio a su ruta de ejecución. Modifique (o cree) un archivo llamado `.profile` en su directorio personal que contenga

```
export PATH=$PATH:~/bin
```

Este archivo debe acabar con una línea en blanco.

Fíjese en que *ruta/hacia* será por lo general `/Applications/`.

2.2 Apoyo respecto de los editores de texto

Existe apoyo por parte de varios editores de texto para LilyPond.

2.2.1 Modo de Emacs

Emacs tiene un '`lilypond-mode`', que proporciona autocompleción de teclado, sangrado, compensación de paréntesis específica de LilyPond y resaltado de sintaxis con colores, útiles combinaciones de teclas para compilar y leer los manuales de LilyPond utilizando Info. Si el '`lilypond-mode`' no está instalado en su sistema, siga leyendo.

Está incluido un modo de Emacs para escribir música y ejecutar LilyPond, en el archivo del código fuente dentro del directorio '`elisp`'. Haga `make install` para instalarlo dentro de *elispdir*. El archivo '`lilypond-init.el`' se debe situar en *load-path*'`/site-start.d/`' o añadirse a su '`~/.emacs`' o '`~/.emacs.el`'.

Como usuario, puede querer añadir su ruta a las fuentes (p.ej. '`~/site-lisp/`') a su *load-path* añadiendo la siguiente línea (modificada) a su '`~/.emacs`'

```
(setq load-path (append (list (expand-file-name "~/site-lisp")) load-path))
```

2.2.2 Modo de Vim

Para VIM se proporciona un '`vimrc`', junto con herramientas de resaltado de sintaxis con colores. Se encuentra incluido un modo de Vim para escribir música y ejecutar LilyPond, dentro del archivo de código fuente, en el directorio `$VIM`.

El tipo de archivo de LilyPond se detecta si el archivo '`~/.vim/filetype.vim`' tiene el siguiente contenido

```
if exists("did_load_filetypes")
  finish
endif
augroup filetypedetect
  au! BufNewFile,BufRead *.ly           setf lilypond
augroup END
```

Incluya esta ruta añadiendo la siguiente línea a su '`~/.vimrc`'

```
set runtimepath+=/usr/local/share/lilypond/${LILYPOND_VERSION}/vim/
```

donde ${LILYPOND_VERSION} es su versión de lilypond. Si Lilypond no se instaló en su momento en '`/usr/local/`', cambie esta ruta de la forma correspondiente.

2.2.3 jEdit

Creado como un «plugin» o añadido para el editor de texto jEdit, LilyPondTool es la herramienta basada en texto más rica en posibilidades para editar partituras de LilyPond. Sus posibilidades incluyen un Asistente de Documentos con apoyo para las letras de las canciones para preparar documentos más fácilmente, y un visor de PDF empotrado con apoyo avanzado para el uso del cursor del ratón. Para ver capturas de pantalla, demostraciones e instrucciones de instalación, visite `http://lilypondtool.organum.hu`

2.2.4 TexShop

El editor TexShop para Mac OS X se puede extender para la ejecución de LilyPond, lilypondbook y convert-ly desde dentro del editor, usando las extensiones disponibles en `http://www.dimi.uniud.it/vitacolo/freesoftware.html`.

2.2.5 TextMate

Hay un tinglado de LilyPond para TextMate. Se puede instalar ejecutando

```
mkdir -p /Library/Application\ Support/TextMate/Bundles
cd /Library/Application\ Support/TextMate/Bundles
svn co http://macromates.com/svn/Bundles/trunk/Bundles/Lilypond.tmbundle/
```

2.2.6 LilyKDE

LilyKDE es un plugin o complemento para el editor de texto de KDE, Kate. Tiene un potente Asistente de Partitura para preparar rápidamente un documento de LilyPond, y un visor de PDF incluido.

LilyKDE puede utilizar Rumor, de manera que se puede introducir la música tocando sobre un teclado MIDI.

Otras posibilidades son el enguionado de la letra y la ejecución de LilyPond sobre varios archivos de una sola vez desde dentro del administrador de archivos de KDE.

2.3 Apuntar y pulsar

«Point and click» (apuntar y pulsar con el ratón) le da la posibilidad de localizar notas del código de entrada pulsando sobre ellas dentro del visor de PDF. Ello facilita encontrar los fragmentos de la entrada que producen algún tipo de error en la partitura.

Cuando esta funcionalidad está activada, LilyPond añade hiper-enlaces al archivo PDF. Estos hiper-enlaces se envían al navegador de web, que a su vez abre un editor de texto con el cursor situado en el lugar correcto.

Para conseguir que esta cadena funcione, tendrá que configurar el visor de PDF de forma que siga los hiper-enlaces usando el guión '`lilypond-invoke-editor`' proporcionado con LilyPond.

Para Xpdf sobre Unix, lo siguiente debe estar presente en '`xpdfrc`'[1]

```
urlCommand     "lilypond-invoke-editor %s"
```

El programa '`lilypond-invoke-editor`' es un pequeño programa de apoyo. Invoca un editor para las URIs especiales de `textedit`, y lanza un navegador de web para el resto. Comprueba la variable de entorno `EDITOR` en busca de los siguientes patrones,

`emacs` esto invoca a

```
emacsclient --no-wait +línea:columna archivo
```

`vim` esto invoca a

```
gvim --remote +:línea:normcarácter archivo
```

`nedit` esto invoca a

```
nc -noask +línea archivo'
```

La variable de entorno `LYEDITOR` se utiliza para sobreescribir esto. Contiene la instrucción de línea de órdenes para abrir el editor, donde `%(archivo)s`, `%(columna)s` y `%(línea)s` se sustituyen por el archivo, columna y línea respectivamente. El ajuste

```
emacsclient --no-wait +%(línea)s:%(columna)s %(archivo)s
```

para `LYEDITOR` equivale a la invocación estándar de emacsclient.

Los enlaces de apuntar y pulsar aumentan significativamente el tamaño de los archivos de salida. Para reducir el tamaño de los archivos PDF y PS, la posibilidad de apuntar y pulsar se puede desactivar escribiendo

```
\pointAndClickOff
```

dentro de un archivo '`.ly`'. Se puede activar explícitamente la posibilidad de apuntar y pulsar con

[1] En Unix, este archivo se encuentra o bien en '`/etc/xpdfrc`' o como '`.xpdfrc`' en su directorio personal.

```
\pointAndClickOn
```

De forma alternativa, puede desactivar la posibilidad de apuntar y pulsar con una opción de la línea de órdenes:

```
lilypond -dno-point-and-click archivo.ly
```

> **Nota:** Recuerde desactivar siempre la posibilidad Apuntar y pulsar en cualquier archivo de LilyPond que vaya a ser distribuido, para evitar incluir información de rutas de archivo relativas a su equipo dentro del archivo .pdf, lo que puede plantear un problema de seguridad.

3 Ejecutar LilyPond

Este capítulo detalla los aspectos técnicos de la ejecución de LilyPond.

3.1 Utilización normal

Casi todos los usuarios ejecutan LilyPond por medio de un interfaz gráfico; consulte Sección "First steps" in *Manual de Aprendizaje* si no lo ha leído aún.

3.2 Utilización desde la línea de órdenes

Esta sección contiene información adicional sobre el uso de LilyPond en la línea de órdenes. Esta forma puede ser preferible para pasarle al programa algunas opciones adicionales. Además, existen algunos programas complementarios 'de apoyo' (como `midi2ly`) que sólo están disponibles en la línea de órdenes.

Al hablar de la 'línea de órdenes', nos referimos a la consola del sistema operativo. Los usuarios de Windows posiblemente estén más familiarizados con los términos 'ventana de MS-DOS' o 'línea de comandos'; Los usuarios de MacOS X puede que estén más familiarizados con los términos 'terminal' o 'consola'. Éstos deberían consultar también el apartado Sección 2.1.1 [Configuración para MacOS X], página 2.

La descripción del uso de esta parte de los sistemas operativos se sale del ámbito de este manuual; le rogamos que consulte otros documentos sobre este tema si no le resulta familiar la línea de órdenes.

3.2.1 Invocar `lilypond`

El ejecutable `lilypond` se puede llamar desde la línea de órdenes de la siguiente manera:

```
lilypond [opción]... archivo...
```

Cuando se invoca con un nombre de archivo sin extensión, se prueba en primer lugar con la extensión '`.ly`'. Para leer la entrada desde stdin, utilice un guión (`-`) en sustitución de *archivo*.

Cuando se procesa '`archivo.ly`', la salida resultante son los archivos '`archivo.ps`' y '`archivo.pdf`'. Se pueden especificar varios archivos; cada uno de ellos se procesará de forma independiente[1].

Si '`archivo.ly`' contiene más de un bloque `\score`, el resto de las partituras se obtendrán como salida en archivos numerados, empezando por '`archivo-1.pdf`'. además, el valor de `output-suffix` (sufijo de salida) se insertará entre el nombre base y el número. Un archivo de entrada que contenga

```
#(define output-suffix "violin")
\book { ... }
#(define output-suffix "cello")
\book { ... }
```

producirá como salida *base*'`-violin.pdf`' y *base*'`-cello-1.pdf`'.

3.2.2 Opciones de la línea de órdenes para `lilypond`

Están contempladas las siguientes opciones:

`-e,--evaluate=`*expresión*

Evaluar la *expresión* de Scheme antes de analizar los archivos '`.ly`'. Se pueden pasar varias opciones `-e`, que se evaluarán en secuencia.

[1] El estado de GUILE no se restablece después de procesar un archivo `.ly`, por lo que debe tener cuidado de no modificar ningún valor predeterminado desde dentro de Scheme.

La expresión se evaluará en el módulo `guile-user`, de manera que si quiere usar definiciones dentro de *expresión*, debe utilizar

```
lilypond -e '(define-public a 42)'
```

en la línea de órdenes, e incluir

```
#(use-modules (guile-user))
```

al principio del archivo `.ly`.

`-f,--format=`*formato*
: Formato de la salida. Como `formato` se puede elegir entre `svg`, `ps`, `pdf` y `png`.

  Ejemplo: `lilypond -fpng` *archivo*`.ly`

`-d,--define-default=`*variable*`=`*valor*
: Establece la opción interna del programa *variable* al valor de Scheme *valor*. Si no se proporciona ningún *valor*, se usa *#t*. Para desactivar una opción se puede anteponer `no-` a la *variable*, p.ej.:

  ```
  -dno-point-and-click
  ```

  es lo mismo que

  ```
  -dpoint-and-click='#f'
  ```

  A continuación veremos algunas opciones interesantes.

  ‘`help`’ La ejecución de `lilypond -dhelp` imprimirá todas las opciones `-d` que están disponibles.

  ‘`paper-size`’
  : Esta opción establece el tamaño predeterminado del papel,

    ```
    -dpaper-size=\"letter\"
    ```

    Observe que la cadena se debe incluir dentro de comillas escapadas (`\"`).

  ‘`safe`’ No confiar en la entrada `.ly`.

  Cuando el proceso de tipografía de LilyPond se encuentra disponible a través de un servidor web, **SE DEBEN** pasar las opciones `--safe` (seguro) o `--jail` (jaula). La opción `--safe` evita que el código de Scheme en línea arme un desastre, por ejemplo

  ```
  #(system "rm -rf /")
  {
    c4^#(ly:export (ly:gulp-file "/etc/passwd"))
  }
  ```

  La opción `-dsafe` funciona evaluando las expresiones en línea de Scheme dentro de un módulo especial seguro. Este módulo seguro deriva del módulo GUILE ‘`safe-r5rs`’, pero añade ciertas funciones del API de LilyPond. Estas funciones se relacionan en ‘`scm/safe-lily.scm`’.

  Además, el modo seguro prohíbe las directivas `\include` e inhabilita el uso de barras invertidas en las cadenas de TEX.

  En el modo seguro, no es posible la importación de variables de LilyPond dentro de Scheme.

  `-dsafe` *no* detecta la sobreutilización de recursos. Aún es posible hacer que el programa se cuelgue indefinidamente, por ejemplo alimentándo el backend con estructuras de datos cíclicas. Por tanto, si se está utilizando LilyPond sobre un servidor web accesible públicamente, el proceso debe limitarse tanto en el uso de CPU como de memoria.

El modo seguro impide que muchos fragmentos útiles de código de LilyPond se puedan compilar. La opción `--jail` es una alternativa más segura, pero su preparación requiere más trabajo.

‘`backend`’ el formato de salida que usar para el back-end o extremo final. Para el `formato` se puede elegir entre

`ps` para PostScript.

Los archivos PostScript incluyen las tipografías TTF, Type1 y OTF. No se seleccionan subconjuntos de estas tipografías. Cuando se usan conjuntos de caracteres orientales, esto puede dar lugar a archivos enormes.

`eps` para obtener PostScript encapsulado. Esto vuelca cada una de las páginas/sistemas como un archivo ‘`EPS`’ distinto, sin tipografías, y como un solo archivo ‘`EPS`’ encuadernado con todas las páginas/sistemas con las tipografías incluidas.

Este modo se usa de forma predeterminada por parte de `lilypond-book`.

`svg` para obtener SVG (gráficos vectoriales escalables). Vuelca cada página como un archivo ‘`SVG`’ distinto, con las tipografías incrustadas. Se necesita un visor de SVG que contemple las tipografías incrustadas, o un visor de SVG que pueda sustituir las tipografías incrustadas por tipografías OTF. Bajo UNIX, puede usar Inkscape (versión 0.42 o posterior), después de copiar las tipografías OTF del directorio de LilyPond (que normalmente es ‘`/usr/share/lilypond/VERSIÓN/fonts/otf/`’) al directorio ‘`~/.fonts/`’.

`scm` para obtener un volcado de las instrucciones internas de dibujo basadas en Scheme, en bruto.

`null` no producir una salida impresa; tiene el mismo efecto que `-dno-print-pages`.

Ejemplo: `lilypond -dbackend=svg` *`archivo`*`.ly`

‘`preview`’ Generar un archivo de salida que contenga solamente los títulos de cabecera y el primer sistema de la primera página.

‘`print-pages`’
Generar las páginas completas, el ajuste predeterminado. `-dno-print-pages` es útil en combinación con `-dpreview`.

`-h,--help`
Mostrar un resumen de las formas de utilización.

`-H,--header=`*`CAMPO`*
Volcar un campo de cabecera al archivo ‘`NOMBREBASE.`*`CAMPO`*’

`--include, -I=`*`directorio`*
Añadir el *directorio* a la ruta de búsqueda de archivos de entrada.

`-i,--init=`*`archivo`*
Establecer el archivo de inicio a *archivo* (predeterminado: ‘`init.ly`’).

`-o,--output=`*ARCHIVO*
: Establecer el nombre del archivo de salida predeterminado a *ARCHIVO*. Se añade el sufijo correspondiente (por ejemplo, `.pdf` para PDF).

`--ps`
: Generar PostScript.

`--png`
: Generar imágenes de las páginas en formato PNG. Esto implica `--ps`. La resolución en PPP de la imagen se puede establecer con

```
-dresolution=110
```

`--pdf`
: Generar PDF. Implica `--ps`.

`-j,--jail=`*usuario*`,`*grupo*`,`*jaula*`,`*directorio*
: Ejecutar `lilypond` en una jaula de chroot.

  La opción `--jail` (jaula) proporciona una alternativa más flexible a la opción `--safe` cuando el proceso de tipografía de LilyPond está disponible a través de un servidor web o cuando LilyPond ejecuta archivos fuente procedentes del exterior.

  La opción `--jail` funciona cambiando la raíz de `lilypond` a *jaula* justo antes de comenzar el proceso de compilación en sí. Entonces se cambian el usuario y el grupo a los que se han dado en la opción, y el directorio actual se cambia a *directorio*. Esta instalación garantiza que no es posible, al menos en teoría, escapar de la jaula. Observe que para que funcione `--jail` se debe ejecutar `lilypond` como root, lo que normalmente se puede hacer de una forma segura utilizando `sudo`.

  La instalación de una jaula es un asunto algo delicado, pues debemos asegurarnos de que LilyPond puede encontrar *dentro de la jaula* todo lo que necesita para poder compilar la fuente. Una configuración típica consta de los siguientes elementos:

  Preparar un sistema de archivos separado
  : Se debe crear un sistema de archivos separado para LilyPond, de forma que se pueda montar con opciones seguras como `noexec`, `nodev` y `nosuid`. De esta forma, es imposible ejecutar programas o escribir directamente a un dispositivo desde LilyPond. Si no quiere crear una partición separada, tan sólo tiene que crear un archivo de un tamaño razonable y usarlo para montar un dispositivo loop. El sistema de archivos separado garantiza también que LilyPond nunca pueda escribir en un espacio mayor del que se le permita.

  Preparar un usuario separado
  : Se debe usar un usuario y grupo separados (digamos `lily/lily`) con bajos privilegios para ejecutar LilyPond dentro de la jaula. Debería existir un solo directorio con permisos de escritura para este usuario, y debe pasarse en el valor *directorio*.

  Preparar la jaula
  : LilyPond necesita leer algunos archivos mientras se ejecuta. Todos estos archivos se deben copiar dentro de la jaula, bajo la misma ruta en que aparecen en el sistema de archivos real de root. Todo el contenido de la instalación de LilyPond (por ejemplo '`/usr/share/lilypond`') se debe copiar.

    Si surgen problemas, la forma más sencilla de rastrearlos es ejecutar LilyPond usando `strace`, lo que le permitirá determinar qué archivos faltan.

  Ejecutar LilyPond
  : Dentro de una jaula montada con `noexec` es imposible ejecutar ningún programa externo. Por tanto, LilyPond se debe ejecutar con un back-

end que no necesite tal programa. Como ya mencionamos, también se debe ejecutar con privilegios del superusuario (que por supuesto perderá inmediatamente), posiblemente usando `sudo`. Es buena idea limitar el número de segundos de tiempo de CPU que LilyPond puede usar (p.ej., usando `ulimit -t`), y, si su sistema operativo lo contempla, el tamaño de la memoria que se puede reservar.

`-v,--version`
: Mostrar la información de la versión.

`-V,--verbose`
: Ser prolijo: mostrar las rutas completas de todos los archivos que se leen, y dar información cronométrica.

`-w,--warranty`
: Mostrar la garantía con que viene GNU LilyPond (¡no viene con **NINGUNA GARANTÍA**!).

3.2.3 Variables de entorno

`lilypond` reconoce las siguientes variables de entorno:

`LILYPOND_DATADIR`
: Especifica un directorio en el que los mensajes de localización y de datos se buscarán de forma predeterminada. El directorio debe contener subdirectorios llamados '`ly/`', '`ps/`', '`tex/`', etc.

`LANG`
: Selecciona el idioma de los mensajes de advertencia.

`LILYPOND_GC_YIELD`
: Con esta variable se puede ajustar la huella y el desempeño de memoria. Es un porcentaje que ajusta el comportamiento de la administración de memoria. Con valores más altos, el programa usa más memoria; con valores más bajos, usa más tiempo de CPU. El valor predeterminado es `70`.

3.3 Mensajes de error

Pueden aparecer distintos mensajes de error al compilar un archivo:

Advertencia
: Algo tiene un aspecto sospechoso. Si estamos pidiendo algo fuera de lo común, entenderemos el mensaje y podremos ignorarlo. Sin embargo, las advertencias suelen indicar que algo va mal con el archivo de entrada.

Error
: Algo va claramente mal. El paso actual de procesamiento (análisis, interpretación o formateo visual) se dará por terminado, pero el siguiente paso se saltará.

Error fatal
: Algo va claramente mal, y LilyPond no puede seguir. Rara vez sucede esto. La causa más frecuente son las tipografías mal instaladas.

Error de Scheme
: Los errores que ocurren al ejecutar código de Schheme se interceptan por parte del intérprete de Scheme. Si se está ejecutando con las opciones `-V` o `--verbose` (prolijo) entonces se imprime una traza de llamadas de la función ofensiva.

Error de programación
: Ha habido algún tipo de inconsistencia interna. Estos mensajes de error están orientados a ayudar a los programadores y a los depuradores. Normalmente se pueden

ignorar. En ocasiones aparecen en cantidades tan grandes que pueden entorpecer la visión de otros mensajes de salida.

Abortado (volcado de core)
: Esto señala un error de programación serio que ha causado la interrupción abrupta del programa. Estos errores se consideran críticos. Si se topa con uno, envíe un informe de fallo.

Se los errores y advertencias se pueden ligar a un punto del archivo de entrada, los mensajes tienen la forma siguiente:

```
archivo:línea:columna: mensaje
línea de entrada problemática
```

Se inserta un salto de línea en la línea problemática para indicar la columna en que se encontró el error. Por ejemplo,

```
prueba.ly:2:19: error: no es una duración: 5
  { c'4 e'
          5 g' }
```

Estas posiciones son la mejor suposición de LilyPond sobre dónde se ha producido el mensaje de error, pero (por su propia naturaleza) las advertencias y errores se producen cuando ocurre algo inesperado. Si no ve un error en la línea que se indica del archivo de entrada, trate de comprobar una o dos líneas por encima de la posición indicada.

3.4 Actualizar ficheros con `convert-ly`

La sintaxis del lenguaje de entrada de LilyPond se modifica de forma habitual para simplificarla o mejorarla de distintas maneras. Como efecto secundario, el intérprete de LilyPond a menudo ya no es compatible con los archivos de entrada antiguos. Para poner remedio a esto se puede utilizar el programa `convert-ly` para manejar casi todos los cambios de sintaxis entre versiones de LilyPond.

Utiliza los enunciados `\version` de los archivos de entrada para detectar el número de versión antiguo. En casi todos los casos, para actualizar el archivo de entrada basta con ejecutar

```
convert-ly -e miarchivo.ly
```

Los usuarios de MacOS X pueden ejecutar esta instrucción bajo el menú `Compilar > Actualizar sintaxis`.

Si no hay cambios en miarchivo.ly y se crea el archivo llamado miarchivo.ly.NEW, entonces miarchivo.ly ya está actualizado.

3.4.1 Opciones de la línea de órdenes para `convert-ly`

`convert-ly` convierte siempre al últimmo cambio de sintaxis que puede manejar. Eesto supone que el número de `\version` que aparece en el archivo convertido suele ser más bajo que la versión del propio programa `convert-ly`.

Para actualizar fragmentos de LilyPond en archivos de texinfo, use

```
convert-ly --from=... --to=... --no-version *.itely
```

Para ver los cambios en la sintaxis de LilyPond entre dos versiones, use

```
convert-ly --from=... --to=... -s
```

Para actualizar muchos archivos de una vez, combine `convert-ly` con las instrucciones estándar de UNIX. Este ejemplo actualiza todos los archivos `.ly` del directorio actual:

```
for f in *.ly; do convert-ly -e $f; done;
```

En general, el programa se invoca de la manera siguiente:

`convert-ly [`*opción*`]... `*archivo*`...`

Se pueden dar las siguientes opciones:

`-e,--edit`
: Hace una edición en línea del archivo de entrada. Sobreescribe a `--output`.

`-f,--from=`*versión_de_origen*
: Establece la versión desde la que convertir. Si no aparece esta opción, `convert-ly` tratará de adivinarla, basándose en el enunciado `\version` del archivo.

`-n,--no-version`
: Normalmente `convert-ly` añade un indicador `\version` a la salida. La especificación de esta opción lo suprime.

`-s, --show-rules`
: Mostrar todas las conversiones conocidas y salir.

`--to=`*versión_final*
: Fijar la versión de destino de la conversión. De forma predeterminada se convierte a la última versión disponible.

`-h, --help`
: Imprimir la ayuda de la utilización.

3.4.2 Problemas con `convert-ly`

No se manejan todos los cambios en el lenguaje. Sólo se puede especificar una opción de salida. La actualización automática de Scheme y los interfaces Scheme de LilyPond es bastante improbable; prepárese para trucar el código de Scheme a mano.

```
Hay algunas cosas que convert-ly no puede manejar.  He aquí una lista
de aquellas limitaciones que han dado lugar a protestas de la
comunidad.

Se ha escogido esta estructura de informe de fallo porque convert-ly
tiene una estructura que no permite implementar de forma progresiva
todos los cambios necesarios.  Así pues esto es sólo una lista de
deseos, y se incluye aquí como referencia.

1.6->2.0:
 No siempre convierte el bajo cifrado correctamente, específicamente cosas como {<
>}.  El comentario de Mats sobre cómo solventar el problema:
   Para poder ejecutar convert-ly
   sobre él, primero sustituí todas las apariciones de '{<' a algo mudo como '{#'
   y de forma similar sustituí '>}' con '&}'.  Después de la conversión, pude
   volver a cambiarlos de '{ #' a '{ <' y de '& }' a '> }'.
 No convierte todos los marcados de texto correctamente.  En sintaxis antigua,
 se podían agrupar varios marcados entre paréntesis, p.ej.
   -#'((bold italic) "cadena")
   Esto se convierte incorrectamente en
   -\markup{{\bold italic} "cadena"}
   en vez del correcto
   -\markup{\bold \italic "cadena"}
2.0->2.2:
 No maneja \partcombine
 No hace \addlyrics => \lyricsto, esto rompe algunas partituras con varias estrofas.
2.0->2.4:
```

```
 \magnify no se cambia por \fontsize.
    - \magnify #m => \fontsize #f, donde f = 6ln(m)/ln(2)
 remove-tag no se cambia.
    - \applyMusic #(remove-tag '. . .) => \keepWithTag #'. . .
 first-page-number no se cambia.
    - first-page-number no => print-first-page-number = ##f
 Los saltos de línea en las cadenas de cabecera no se convierten.
    - \\\\  como salto de línea en las cadenas de \header  => \markup \center-align <
      "Primera línea" "Segunda línea" >
 Los terminadores de crescendo y decrescendo no se convierten.
    - \rced => \!
    - \rc => \!
2.2->2.4:
 \turnOff (usado en \set Staff.VoltaBracket = \turnOff) no se convierte
adecuadamente.
2.4.2->2.5.9
 \markup{ \center-align <{ ... }> } se tendría que convertir en:
 \markup{ \center-align {\line { ... }} }
 pero ahora, falta el \line.
2.4->2.6
 Los caracteres especiales de LaTeX como $~$ en el texto no se convierten a UTF8.
2.8
 \score{} ahora debe empezar con una expresión musical.  Cualquier otra cosa
 (en particular, \header{}) debe ir después de la música.
```

3.5 Informar de fallos

Si tiene una entrada que produce una interrupción abrupta o una salida errónea, entonces eso es un bug (fallo). Hay una lista de los fallos actuales en nuestro rastreador de fallos de Google Code:

http://code.google.com/p/lilypond/issues/list

Si descubre un error que no está en la lista, le rogramos que envíe un informe del fallo siguiendo las instrucciones que aparecen en

http://lilypond.org/web/devel/participating/bugs

Le rogamos, asimismo, que para los informes prepare y envíe ejemplos mínimos de los fallos. No tenemos los recursos para investigar informes que no sean lo más pequeños posible.

4 lilypond-book: integrar texto y música

Si quiere añadir imágenes de música a un documento, puede hacerlo simplemente de la forma en que lo haría con otros tipos de imágenes. Las imágenes se crean por separado, dando como resultado una salida PostScript o imágenes PNG, y luego se incluyen en un documento de LaTeX o de HTML.

`lilypond-book` ofrece una manera de automatizar este proceso: este programa extrae los fragmentos de música del documento, ejecuta `lilypond` sobre cada uno de ellos, y devuelve como salida el documento con la música sustituida por las imágenes. Las definiciones de ancho de línea y tamaño de letra de la música se ajustan de forma que coincidan con los ajustes de su documento.

Es un programa distinto a `lilypond` propiamente dicho, y se ejecuta sobre la línea de órdenes; para ver más información, consulte Sección 3.2 [Utilización desde la línea de órdenes], página 6. Si tiene MacOS 10.3 o 10.4 y experimenta algún problema al ajecutar `lilypond-book`, consulte Sección 2.1.1 [Configuración para MacOS X], página 2.

Este procedimiento se puede aplicar a documentos de LaTeX, HTML, Texinfo o DocBook.

4.1 Un ejemplo de documento musicológico

Ciertos textos contienen ejemplos musicales. Son tratados musicales, cancioneros o manuales como este mismo. Estos textos se pueden hacer a mano, importando simplemente una imagen en formato PostScript en el editor de textos. Sin embargo, hay un procedimiento automático para reducir la carga de trabajo que esto implica los documentos de HTML, LaTeX, Texinfo y DocBook.

Un guión ejecutable llamado `lilypond-book` extrae los fragmentos de música, les da formato y vuelve a poner en su lugar la partitura resultante. A continuación presentamos un pequeño ejemplo de su utilización con LaTeX. El ejemplo contiene también texto explicativo, por lo que no vamos a comentarlo posteriormente.

Entrada

```
\documentclass[a4paper]{article}

\begin{document}

Los documentos para \verb+lilypond-book+ pueden mezclar libremente
música y texto. Por ejemplo:

\begin{lilypond}
\relative c' {
  c2 g'2 \times 2/3 { f8 e d } c'2 g4
}
\end{lilypond}

Las opciones se escriben entre corchetes.

\begin[fragment,quote,staffsize=26,verbatim]{lilypond}
  c'4 f16
\end{lilypond}

Los ejemplos grandes se pueden grabar en archivos separados e
introducirse con \verb+\lilypondfile+.
```

```
\lilypondfile[quote,noindent]{screech-boink.ly}

(Si es necesario, sustituya screech-boink.ly por cualquier archivo .ly
situado en el mismo directorio que este archivo.)

\end{document}
```

Procesado

Guarde el código anterior como un archivo llamado '`lilybook.lytex`', y luego ejecute en un terminal:

```
lilypond-book --output=out --pdf lilybook.lytex
lilypond-book (GNU LilyPond) 2.12.1

Leyendo lilybook.lytex...
..montañas de mensajes suprimidos..
Compilando lilybook.tex...
cd out
pdflatex lilybook
..montañas de mensajes suprimidos..
xpdf lilybook
(sustituya xpdf por su visor de PDF favorito)
```
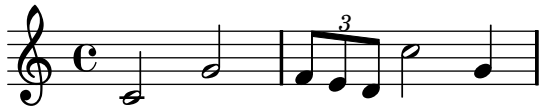
La ejecución de `lilypond-book` y `latex` crea un gran número de archivos temporales, que podrían abarrotar el directorio de trabajo. Para poner remedio a esto utilice la opción `--output=directorio`. Creará los archivos en un subdirectorio aparte '`directorio`'.

Finalmente el resultado del ejemplo de LaTeX que acabamos de mostrar[1]. Así acaba la sección del tutorial.

[1] Este tutorial se procesa con Texinfo, por lo que el ejemplo presenta un resultado en la disposición ligeramente distinto.
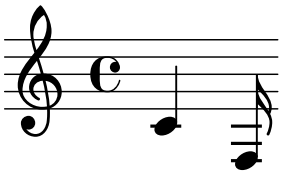
Salida

Los documentos para `lilypond-book` pueden mezclar libremente música y texto. Por ejemplo:



Las opciones se escriben entre corchetes.

```
c'4 f16
```

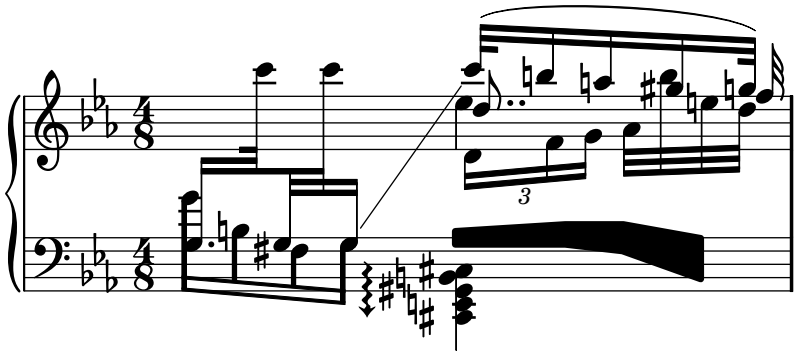


Los ejemplos grandes se pueden grabar en archivos separados e introducirse con `\lilypondfile`.

Screech and boink

Random complex notation

Han-Wen Nienhuys



4.2 Integrar música y texto

Aquí vamos a explicar cómo integrar LilyPond con algunos otros formatos de salida.

4.2.1 LaTeX

LaTeX es el estándar de facto para la publicación en el mundo de las ciencias exactas. Está construido encima del motor de composición tipográfica TeX, proporcionando la tipografía de mejor calidad que existe.

Consulte *The Not So Short Introduction to LaTeX* (Introducción no tan breve a LaTeX) para ver una panorámica sobre cómo usar LaTeX.

La música se introduce usando

```
\begin[las,opciones,van,aquí]{lilypond}
  EL CÓDIGO DE LILYPOND
\end{lilypond}
```

o bien

```
\lilypondfile[las,opciones,van,aquí]{archivo}
```

o bien

```
\lilypond{ EL CÓDIGO DE LILYPOND }
```
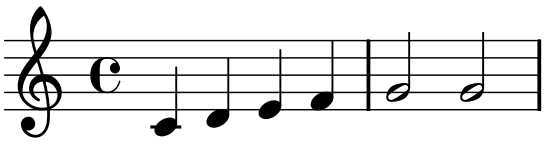
De forma adicional, `\lilypondversion` imprime la versión actual de lilypond.

La ejecución de `lilypond-book` deja como resultado un archivo que se puede procesar posteriormente con LaTeX.

A continuación mostramos algunos ejemplos. El entorno `lilypond`

```
\begin[quote,fragment,staffsize=26]{lilypond}
  c' d' e' f' g'2 g'2
\end{lilypond}
```

produce



La versión corta

```
\lilypond[quote,fragment,staffsize=11]{<c' e' g'>}
```

produce



Por el momento no es posible incluir llaves `{` o `}` dentro de `\lilypond{}`, así que esta instrucción solamente es útil con la opción `fragment`.

El ancho predeterminado de las líneas de música se ajusta mediante el examen de las instrucciones del preámbulo del documento, la parte del documento que está antes de `\begin{document}`. La instrucción `lilypond-book` los envía a LaTeX para averiguar la anchura del texto. El ancho de la línea para los fragmentos de música se ajusta entonces al ancho del texto. Observe que este algoritmo heurístico puede fácilmente fallar; en estos casos es necesario usar la opción `line-width` del fragmento de música.

Cada fragmento ejecutará los macros siguientes si han sido definidos por el usuario:

bullet `\preLilyPondExample` que se llama antes de la música,

bullet `\postLilyPondExample` que se llama después de la música,

bullet `\betweenLilyPondSystem[1]` se llama entre los sistemas si `lilypond-book` ha dividido el fragmento en varios archivos PostScript. Se debe definir de forma que tome un parámetro y recibirá el número de archivos ya incluidos dentro del fragmento actual. La acción predeterminada es simplemente insertar un `\linebreak`.

Fragmentos de código seleccionados

A veces es útil mostrar elementos de música (como ligaduras) como si continuasen más allá del final del fragmento. Esto se puede hacer dividiendo el pentagrama y suprimiendo la inclusión del resto de la salida de LilyPond.

En LaTeX, defina `\betweenLilyPondSystem` de tal forma que la la inclusión de otros sistemas se dé por terminada una vez que se ha alcanzado el número deseado de sistemas requeridos. Puesto que `\betweenLilypondSystem` se llama en primer lugar *después* del primer sistema, incluir solamente el primer sistema es algo trivial.

```
\def\betweenLilyPondSystem#1{\endinput}

\begin[fragment]{lilypond}
  c'1\( e'( c'~ \break c' d) e f\)
\end{lilypond}
```

Si se necesita un mayor número de sistemas, se tiene que usar un condicional de TeX antes del `\endinput`. En este ejemplo, sustituya el '2' por el número de sistemas que quiere en la salida:

```
\def\betweenLilyPondSystem#1{
    \ifnum##1<2\else\endinput\fi
}
```

Recuerde que la definición de `\betweenLilyPondSystem` es efectiva hasta que TeX abandona el grupo actual (como el entorno LaTeX) o se sobreescribe por otra definición (lo que casi siempre es por el resto del documento). Para reponer la definición, escriba

```
\let\betweenLilyPondSystem\undefined
```

dentro de la fuente de LaTeX.

Se puede simplificar esto definiendo un macro de TeX:

```
\def\onlyFirstNSystems#1{
    \def\betweenLilyPondSystem##1{\ifnum##1<#1\else\endinput\fi}
}
```

y luego diciendo solamente cuántos sistemas quiere antes de cada fragmento:

```
\onlyFirstNSystems{3}
\begin{lilypond}...\end{lilypond}
\onlyFirstNSystems{1}
\begin{lilypond}...\end{lilypond}
```

Véase también

Hay opciones de línea de órdenes específicas de `lilypond-book` y otros detalles que conocer para procesar documentos deLaTeX véase Sección 4.4 [Invocar lilypond-book], página 23.

4.2.2 Texinfo

Texinfo es el formato estándar para la documentación del proyecto GNU. Este mismo manual es un ejemplo de documento Texinfo. Las versiones HTML, PDF e Info del manual se hacen a partir del documento Texinfo.

En el archivo de entrada, la música se especifica con

```
@lilypond[las,opciones,van,aquí]
  EL CÓDIGO DE LILYPOND
@end lilypond
```

o bien

```
@lilypond[las,opciones,van,aquí]{ EL CÓDIGO DE LILYPOND }
```

o bien

```
@lilypondfile[las,opciones,van,aquí]{archivo}
```
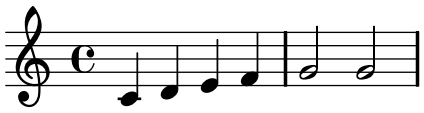
De forma adicional, `@lilypondversion` imprime la versión actual de lilypond.

Cuando se ejecuta `lilypond-book` sobre el archivo, se obtiene como resultado un archivo Texinfo (con la extensión '`.texi`') que contiene etiquetas `@image` para el HTML, Info y la salida impresa. `lilypond-book` genera imágenes de la música en formatos EPS y PDF para usarlos en la salida impresa, y en formato PNG para usarlos en las salidas HTML e Info.

Aquí podemos ver dos ejemplos sencillos. Un entorno `lilypond`

```
@lilypond[fragment]
c' d' e' f' g'2 g'
@end lilypond
```

produce



La versión corta

```
@lilypond[fragment,staffsize=11]{<c' e' g'>}
```

produce
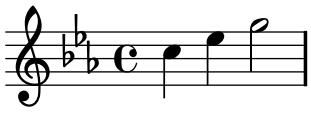


No como LaTeX, `@lilypond{...}` no genera una imagen en línea. Siempre consiste en un párrafo para ella sola.

4.2.3 HTML

La música se introduce usando

```
<lilypond fragment relative=2>
\key c \minor c4 es g2
</lilypond>
```

`lilypond-book` entonces produce un archivo HTML con las etiquetas de imagen adecuadas para los fragmentos de música:



Para imágenes en línea, utilice `<lilypond ... />`, donde las opciones están separadas de la música por el símbolo de dos puntos, por ejemplo

```
Algo de música dentro de <lilypond relative=2: a b c/> una línea de texto.
```

Para incluir archivos externos, escriba

```
<lilypondfile opción1 opción2 ...>archivo</lilypondfile>
```

De forma adicional, `<lilypondversion/>` imprime la versión actual de lilypond.

4.2.4 DocBook

Para insertad fragmentos de LilyPond es bueno tratar de mantener la conformidad del documento de DocBook, permitiendo así el uso de editores de DocBook, validación, etc. Así pues, no usamos etiquetas personalizadas, sólo especificamos una convención basada en los elementos estándar de DocBook.

Convenciones usuales

Para insertar toda clase de fragmentos utilizamos los elementos `mediaobject` y `inlinemediaobject`, de forma que nuestros fragmentos puedan ser formateados en línea o no en línea. Las opciones de formateo del fragmento se escriben siempre dentro de la propiedad `role` del elemento más interno (véanse las secciones siguientes). Las etiquetas se eligen de forma que permitan a los editores de DocBook formatear el contenido satisfactoriamente. Los archivos de DocBook que se van a procesar con `lilypond-book` deben tener la extensión '`.lyxml`'.

Incluir un archivo de LilyPond

Este es el caso más sencillo. Debemos usar la extensión '`.ly`' para el archivo incluido, e insertarlo como un `imageobject` estándar, con la estructura siguiente:

```
<mediaobject>
  <imageobject>
    <imagedata fileref="music1.ly" role="printfilename" />
  </imageobject>
</mediaobject>
```

Observe que puede usar mediaobject o inlinemediaobject como el elemento más externo, a elección suya.

Incluir código de LilyPond

Se puede incluir código de LilyPond mediante la utilización de un elemento `programlisting`, en que el lenguaje se establece como `lilypond` con la estructura siguiente:

```
<inlinemediaobject>
  <textobject>
    <programlisting language="lilypond" role="fragment verbatim staffsize=16 ragged-rig
\context Staff \with {
  \remove Time_signature_engraver
  \remove Clef_engraver}
  { c4( fis) }
    </programlisting>
  </textobject>
</inlinemediaobject>
```

Como puede ver, el elemento más externo es un `mediaobject` o un `inlinemediaobject`, y hay un elemento `textobject` que lleva el `programlisting` en su interior.

Procesar el documento de DocBook

Al ejecutar `lilypond-book` sobre el archivo '`.lyxml`' se creará un documento de DocBook válido que se puede procesar posteriormente con la extensión '`.xml`'. Si usa dblatex, creará un archivo PDF a partir de este documento automáticamente. Para la generación de HTML (HTML Help, JavaHelp, etc.) puede usar las hojas de estilo oficiales XSL de DocBook, aunque es posible que tenga que aplicarles algún tipo de personalización.

4.3 Opciones de fragmentos de música

Durante los próximos párrafos, una 'instrucción de LilyPond' se refiere a cualquier instrucción descrita en las secciones anteriores que se maneja por parte de `lilypond-book` para que produzca un fragmento de música. Por simplicidad, las instrucciones de LilyPond solamente se muestran en la sintaxis de LaTeX.

Observe que la cadena de opciones se analiza de izquierda a derecha; si una opción aparece varias veces, se toma la última solamente.

Están disponibles las siguientes opciones para las instrucciones de LilyPond:

`staffsize=`*altura*
: Establecer la altura del pentagrama como *altura*, medida en puntos.

`ragged-right`
: Producir líneas no justificadas por la derecha y con espaciado natural, es decir, se añade `ragged-right = ##t` al fragmento de LilyPond. Esta es la opción predeterminada para la instrucción `\lilypond{}` si no está presente la opción `line-width`. También es la opción predeterminada para el entorno `lilypond` si está establecida la opción `fragment`, y no se especifica la anchura de la línea explícitamente.

`noragged-right`
: Para fragmentos de una sola línea, permitir que la longitud del pentagrama se amplíe hasta igualar la anchura de la línea, es decir, se añade `ragged-right = ##f` al fragmento de LilyPond.

`line-width`
`line-width=`*tamaño*`\`*unidades*
: Establecer el ancho de línea como *tamaño*, utilizando *unidades* como unidad. *unidades* es una de las siguientes cadenas: `cm`, `mm`, `in` o `pt`. Esta opción afecta a la salida de LilyPond (esto es, a la longitud del pentagrama del fragmento musical), no al formato del texto.

  Si se usa sin ningún argumento, se establece el ancho de la línea a un valor predeterminado (calculado con un algoritmo heurístico).

  Si no se da ninguna opción `line-width`, `lilypond-book` trata de adivinar un valor predeterminado para los entornos `lilypond` que no usan la opción `ragged-right`.

`notime`
: No imprimir la indicación de compás, y desactivar las indicaciones temporales de la música (indicación del compás y líneas divisorias).

`fragment`
: Hacer que `lilypond-book` añada algunos códigos necesarios para que podamos escribir simplemente, por ejemplo,

  ```
  c'4
  ```

  sin `\layout`, `\score`, etc.

`nofragment`
: No añadir escribir el código adicional que completa la sintaxis de LilyPond en los fragmentos de música. Al ser la opción predeterminada, `nofragment` normalmente es redundante.

`indent=`*tamaño*`\`*unidades*
: Establecer el sangrado del primer sistema de pentagramas como *tamaño*, utilizando *unidades* como unidad. *unidades* es una de las siguientes cadenas: `cm`, `mm`, `in` o `pt`. Esta opción afecta a LilyPond, no al formato del texto.

`noindent`
: Establecer el sangrado del primer sistema de la música como cero. Esta opción afecta a LilyPond, no al formato del texto. Puesto que el valor predeterminado es que no haya ningún sangrado, `noindent` normalmente es redundante.

`quote` Reducir la longitud de la línea de un fragmento musical en $2 * 0.4$ in (pulgadas) y colocar la salida dentro de un bloque de cita (quotation). El valor de '0.4 in' se puede controlar con la opción `exampleindent`.

`exampleindent`
: Establecer la longitud del sangrado que la opción `quote` aplica al fragmento musical.

`relative`
`relative=n`
: Usar el modo de octava relativa. De forma predeterminada, las notas se especifican con relación al Do central. El argumento entero opcional especifica la octava de la nota inicial, donde el valor predeterminado `1` es el Do central. La opción `relative` sólo funciona cuando está establecida la opción `fragment`, de manera que `fragment` viene implicada automáticamente por `relative`, independientemenre de la presencia de `fragment` o de `nofragment` en la fuente.

LilyPond utiliza también `lilypond-book` para producir su propia documentación. Para hacerlo, están a nuestra disposición ciertas opciones algo esotéricas para los fragmentos musicales.

`verbatim` El argumento de una instrucción de LilyPond se copia al archivo de salida y se incluye dentro de un bloque «verbatim» o preformateado, seguido del texto que se escriba con la opción `intertext` (que no funciona aún); después se imprime la música en sí. Esta opción no funciona bien con `\lilypond{}` si forma parte de un párrafo.

Si se usa la opción `verbatim` dentro de una instrucción `lilypondfile`, es posible incluir con estilo preformateado sólo una parte del archivo fuente. Si el archivo de código fuente contiene un comentario que contiene '`begin verbatim`' (sin las comillas), la cita del bloque de estilo preformateado empezará después de la última vez que aparezca este comentario; de forma similar, la cita del bloque preformateado se detendrá justo antes de la primera vez que aparezca un comentario que contenga '`end verbatim`', si lo hay. En el siguiente ejemplo de código fuente, la música se interpreta en el modo relativo, pero la cita preformateada no presentará el bloque `relative`, es decir

```
\relative c' { % begin verbatim
  c4 e2 g4
  f2 e % end verbatim
}
```

se imprimirá como un bloque preformateado como

```
  c4 e2 g4
  f2 e
```

Si queremos traducir los comentarios y los nombres de variable en la salida literal pero no en el código fuente, podemos establecer el valor de la variable de entorno `LYDOC_LOCALEDIR` a la ruta de un directorio; este directorio debe contener un árbol de catálogos de mensajes '`.mo`' con `lilypond-doc` como dominio.

`addversion`
: (Sólo para la salida de Texinfo.) Anteponer la línea `\version @w{"@version{}"}` a la salida de `verbatim`.

`texidoc` (Sólo para la salida de Texinfo.) Si se llama a `lilypond` con la opción '`--header=texidoc`', y el archivo que se procesa se llama '`fulanito.ly`', crea un archivo '`fulanito.texidoc`' si existe un campo `texidoc` dentro del bloque `\header` de cabecera. La opción `texidoc` hace que `lilypond-book` incluya estos archivos, añadiendo su contenido como un bloque de documentación inmediatamente antes del fragmento musical.

Suponiendo que el archivo ‘fulanito.ly’ contiene

```
\header {
  texidoc = "Este archivo es un ejemplo de una sola nota."
}
{ c'4 }
```

y que tenemos lo siguiente en nuestro documento de Texinfo ‘prueba.texinfo’

```
@lilypondfile[texidoc]{fulanito.ly}
```

la siguiente orden da como salida el resultado esperado:

```
lilypond-book --pdf --process="lilypond \
  -dbackend=eps --header=texidoc" test.texinfo
```

La mayoría de los documentos de prueba de LilyPond (en el directorio ‘input’ de la distribución) son pequeños archivos ‘.ly’ que tienen exactamente este aspecto.

Por motivos de localización de idioma, si el documento de Texinfo contiene `@documentlanguage` *LANG* y la cabecera de ‘loquesea.ly’ contiene un campo `texidoc`*LANG*, y `lilypond` se ejecuta con ‘`--header=texidoc`*LANG*’, entonces se incluirá ‘`loquesea.texidoc`*LANG*’ en lugar de ‘loquesea.texidoc’.

`lilyquote`
: (Sólo para la salida de Texinfo.) Esta opción es similar a quote, pero se pone dentro del bloque de cita solamente el fragmento de música (y el bloque preformateado que se da en la opción `verbatim`). Esta opción es útil si queremos citar (`quote`) el fragmento musical pero no el bloque de documentación `texidoc`.

`doctitle`
: (Sólo para la salida de Texinfo.) Esta opción funciona de forma parecida a la opción `texidoc`: si `lilypond` se llama con la opción ‘`--header=doctitle`’, y el archivo que procesar se llama ‘loquesea.ly’ y contiene un campo `doctitle` en el bloque `\header`, crea un archivo ‘loquesea.doctitle’. Cuando se usa la opción `doctitle`, el contenido de ‘loquesea.doctitle’, que debería ser una línea única de *texto*, se inserta en el documento de Texinfo como `@lydoctitle` *texto*. `@lydoctitle` debe ser un macro definido en el documento de Texinfo. La misma indicación referida al procesado de `texidoc` con idiomas localizados se aplica a `doctitle`.

`nogettext`
: (Sólo para la salida de Texinfo.) No traducir los comentarios y nombres de variable en el fragmento de código literal citado.

`printfilename`
: Si un archivo de entrada de LilyPond se incluye con `\lilypondfile`, imprimir el nombre del archivo inmediatamente antes del fragmento musical. Para la salida HTML, esto es un enlace. Sólo se imprime el nombre base del archivo, es decir, se elimina la parte del directorio de la ruta del archivo.

`fontload`
: Esta opción incluye las fuentes tipográficas en todos los archivos EPS generados para este fragmento. Se debe usar si el fragmento usa cualquier tipografía que LaTeX no es capaz de encontrar por sí solo.

4.4 Invocar `lilypond-book`

`lilypond-book` produce un archivo con una de las siguientes extensiones: ‘.tex’, ‘.texi’, ‘.html’ o ‘.xml’, dependiendo del formato de salida. Todos los archivos ‘.tex’, ‘.texi’ y ‘.xml’ necesitan un procesado posterior.

Instrucciones específicas de formato

LaTeX

Hay dos formas de procesar el documento en LaTeX para su impresión o publicación: hacer un archivo PDF directamente con PDFLaTeX, o generar un archivo PostScript con LaTeX a través de un traductor de DVI a PostScript como `dvips`. la primera forma es más sencilla y es la que se recomienda[1], y cualquiera que sea el método que utilice, podrá convertir fácilmente entre PostScript y PDF con herramientas como `ps2pdf` y `pdf2ps` que vienen incluidas con Ghostscript.

Para producir un archivo PDF por medio de PDFLaTeX, utilice

```
lilypond-book --pdf miarchivo.pdftex
pdflatex miarchivo.tex
```

Para producir una salida PDF por medio de LaTeX/`dvips`/`ps2pdf`, debe hacer

```
lilypond-book miarchivo.lytex
latex miarchivo.tex
dvips -Ppdf miarchivo.dvi
ps2pdf miarchivo.ps
```

El archivo '`.dvi`' creado por este proceso no contiene las cabezas de las notas. Esto es normal; si sigue las instrucciones, las cabezas aparecerán en los archivos '`.ps`' y '`.pdf`'.

La ejecución de `dvips` puede dar como resultado algunas advertencias sobre las fuentes tipográficas; son inocuas y se pueden ignorar. Si está ejecutando `latex` en modo de dos columnas, recuerde añadir `-t landscape` a las opciones de `dvips`.

Texinfo

Para producir un documento de Texinfo (en cualquier formato de salida), siga el procedimiento normal para Texinfo, esto es: o bien llame a `texi2pdf` o a `texi2dvi` o a `makeinfo`, según el formato de la salida que quiera crear. Consulte la documentación de Texinfo para ver más detalles.

Opciones de la línea de órdenes

`lilypond-book` acepta las siguientes opciones de la línea de órdenes:

`-f` *formato*
`--format=`*formato*
: Especificar el tipo del documento que se va a procesar: `html`, `latex`, `texi` (predeterminado) o `docbook`. Si falta esta opción, `lilypond-book` tratará de detectar el formato automáticamente, véase Sección 4.5 [Extensiones de nombres de archivo], página 26. Por el momento, `texi` es lo mismo que `texi-html`.

`-F` *filtro*
`--filter=`*filtro*
: Conducir los fragmentos a través de *filter* por medio de una tubería. `lilypond-book` no obedecerá –filter y –process al mismo tiempo. Por ejemplo,

```
lilypond-book --filter='convert-ly --from=2.0.0 -' mi-libro.tely
```

`-h`
`--help`
: Imprimir un breve mensaje de ayuda.

`-I` *directorio*
`--include=`*directorio*
: Añadir *directorio* a la ruta de inclusión. `lilypond-book` busca también los fragmentos ya compilados en la ruta de inclusión, y no los vuelve a escribir en el directorio

[1] Observe que PDFLaTeX y LaTeX podrían no ser utilizables para compilar cualquier documento LaTeX, y es por lo que explicamos las dos formas.

de salida, así que en ciertos casos es necesario invocar instrucciones de procesado posteriores como `makeinfo` o `latex` con las mismas opciones `-I` *`directorio`*.

`-o` *`directorio`*
`--output=`*`directorio`*
: Colocar los archivos generados en el *directorio*. La ejecución de `lilypond-book` genera montañas de pequeños archivos que luego procesará LilyPond. Para evitar toda esta parafernalia en el mismo directorio que la fuente, utilice la opción '`--output`', y cambie a este directorio antes de ejecutar `latex` o `makeinfo`.

```
lilypond-book --output=out miarchivo.lytex
cd out
...
```

`--skip-lily-check`
: Evitar el fracaso si no se encuentra ninguna salida de lilypond. Se usa para la documentación de LilyPond en formato Info sin imágenes.

`--skip-png-check`
: Evitar el fracaso si no se encuentran las imágenes PNG de los archivos EPS. Se usa para la documentación de LilyPond en formato Info sin imágenes.

`--lily-output-dir=`*`directorio`*
: Escribir archivos lily-XXX en el directorio *directorio*, enlazar en el directorio de `--output`. Use esta opción para ahorrar tiempo de construcción para documentos de distintos directorios que comparten muchos fragmentos idénticos de código.

`--info-images-dir=`*`directorio`*
: Dar formato a la salida de Texinfo de manera que Info busque las imágenes de música en *directorio*.

`--latex-program=`*`prog`*
: Ejecutar el propgrama `prog` en vez de `latex`. Esto es útil si nuestro documento se procesa con `xelatex`, por ejemplo.

`--left-padding=`*`cantidad`*
: Rellenar las cajas EPS en esta medida, alrededor. *cantidad* se mide en milímetros, y es 3.0 como valor predeterminado. Esta opción se debe usar si las líneas de música están muy pegadas al margen derecho.

  El ancho de un sistema que está muy ajustado dentro de su rectángulo puede variar, debido a los elementos de notación que están pegados al margen izquierdo, como los números de compás y el nombre del instrumento. Esta opción acorta todas las líneas y las mueve a la derecha en la misma medida.

`-P` *`instrucción`*
`--process=`*`instrucción`*
: Procesar los fragmentos de LilyPond utilizando *instrucción*. La instrucción predeterminada es `lilypond`. `lilypond-book` no obedecerá a `--filter` y a `--process` al mismo tiempo.

`--pdf` Crear archivos PDF para su uso con PDFLaTeX.

`--psfonts`
: Extraer todas las tipografías PostScript en '*`archivo`*`.psfonts`' para dvips. Esto es necesario para `dvips -h` *`archivo`*`.psfonts`.

`-V`
`--verbose`
: Ser prolijo.

`-v`
`--version`
: Imprimir la información de la versión.

Advertencias y problemas conocidos

La instrucción de Texinfo `@pagesizes` no se interpreta. De forma similar, las instrucciones de LaTeX que cambian los márgenes y anchos de línea después del preámbulo se ignoran.

Sólo se procesa el primer `\score` de un bloque LilyPond.

4.5 Extensiones de nombres de archivo

Puede usar cualquier extensión para el nombre del archivo de entrada, pero si no usa la extensión recomendada para un formato en particular tendrá que especificar manualmente el formato de salida; para ver más detalles, consulte Sección 4.4 [Invocar lilypond-book], página 23. En caso contrario, `lilypond-book` selecciona automáticamente el formato de salida basándose en la extensión del nombre del archivo de entrada.

extensión	**formato de salida**
'`.html`'	HTML
'`.itely`'	Texinfo
'`.latex`'	LaTeX
'`.lytex`'	LaTeX
'`.lyxml`'	DocBook
'`.tely`'	Texinfo
'`.tex`'	LaTeX
'`.texi`'	Texinfo
'`.texinfo`'	Texinfo
'`.xml`'	HTML

Si usa la misma extensión para el archivo de entrada que la que usa `lilypond-book` para el archivo de salida, y si el archivo de entrada está en el mismo directorio que el directorio de trabajo de `lilypond-book`, debe usar la opción `--output` para que funcione `lilypond-book`, pues en caso contrario saldrá con un mensaje de error como "La salida sobreescribirá al archivo de entrada".

4.6 Otros métodos para mezclar texto y música

Esta sección presenta métodos para integrar texto y música distintos del método automatizado con `lilypond-book`.

Muchas citas de una partitura extensa

Si tiene que citar muchos fragmentos extraídos de una partitura grade, puede también usar la capacidad de recorte de sistemas, véase Sección "Extracción de fragmentos de música" in *Referencia de la Notación*.

Insertar la salida de LilyPond dentro de OpenOffice.org

Se puede añadir notación de LilyPond a los documentos de OpenOffice.org con OOoLilyPond.

Insertar la salida de LilyPond dentro de otros programas

Para insertar la salida de LilyPond dentro de otros programas, use `lilypond` en vez de `lilypond-book`. Cada ejemplo debe crearse individualmente y añadirse al documento; consulte la documentación del programa correspondiente. La mayoría de los programas podrán insertar la salida de LilyPond en los formatos '`PNG`', '`EPS`' o '`PDF`'.

Para reducir el espacio vacío alrededor de la partitura de LilyPond, utilice las siguientes opciones:

```
\paper{
  indent=0\mm
  line-width=120\mm
  oddFooterMarkup=##f
  oddHeaderMarkup=##f
  bookTitleMarkup = ##f
  scoreTitleMarkup = ##f
}

{ c1 }
```

Para obtener un archivo 'EPS' que sea útil, utilice

```
lilypond -dbackend=eps -dno-gs-load-fonts -dinclude-eps-fonts miarchivo.ly
```

'PNG':

```
lilypond -dbackend=eps -dno-gs-load-fonts -dinclude-eps-fonts --png miarchivo.ly
```

5 Conversión desde otros formatos

También se puede escribir la música a través de su importación desde otros formatos. Este capítulo trata de documentar las herramientas incluidas en la dirstribución que permiten hacerlo. Existen otras herramientas que producen código de entrada de LilyPond, como por ejemplo secuenciadores con interfaz gráfico y comvertidores de XML. Consulte el website para ver más detalles.

Son programas distintos a `lilypond` propiamente dicho, y se ejecutan desde la línea de órdenes; consulte Sección 3.2 [Utilización desde la línea de órdenes], página 6 para ver más información. Si tiene MacOS 10.3 o 10.4 y tiene problemas para ejecutar alguno de estos guiones, p.ej. `convert-ly`, consulte Sección 2.1.1 [Configuración para MacOS X], página 2.

Advertencias y problemas conocidos

Por desgracia no disponemos de los recursos necesarios para poder mantener estos programas; le rogramos que los tome "tal cual están". Se agradecerá el envío de parches correctores, pero los informes de fallo casi con certeza no se resolverán a medio plazo.

5.1 Invocar `midi2ly`

`midi2ly` traduce un archivo MIDI de tipo 1 a un acrhivo de código fuente de LilyPond.

El MIDI (Music Instrument Digital Interface, Interfase Digital para Instrumentos Musicales) es un estándar para instrumentos digitales: especifica la interconexión física, un protocolo en serie y un formato de archivo. El formato de archivo MIDI es un formato estándar de facto para exportar música de otros programas, por lo que esta posibilidad puede ser de utilidad al importar archivos de un programa que tiene un convertidor para un formato directo.

`midi2ly` convierte las pistas en contextos de Sección "Staff" in *Referencia de Funcionamiento Interno*) y los canales en contextos de Sección "Voice" in *Referencia de Funcionamiento Interno*. Se utiliza el modo relativo para las alturas, y las duraciones se escriben solamente cuando es necesario.

Es posible grabar un archivo MIDI usando un teclado digital y convertirlo después a '`.ly`'. Sin embargo, los intérpretes humanos no son lo suficientemente exactos desde el punto de vista rítmico como para hacer que sea trivial la conversión de MIDI a LY. Si se invoca con las opciones de cuantización (`-s` y `-d`), `midi2ly` intenta compensar los errores de medida, pero no lo hace muy bien. Por ello, no se recomienda usar `midi2ly` para archivos midi generados por seres humanos.

Se invoca a partir de la línea de órdenes como sigue:

```
midi2ly [opción]... archivo_midi
```

Observe que al decir 'línea de órdenes', queremos decir la línea de órdenes del sistema operativo. Consulte Capítulo 5 [Conversión desde otros formatos], página 28 para ver más información sobre esto.

`midi2ly` contempla las siguientes opciones:

`-a, --absolute-pitches`
: Imprimir alturas absoultas.

`-d, --duration-quant=`*DURACIÓN*
: Cuantizar las duraciones a *DURACIÓN*.

`-e, --explicit-durations`
: Imprimir valores de figura explícitos.

`-h,--help`
: Mostrar un resumen de las instrucciones de utilización.

`-k, --key=`*`alteración`*`[:`*`minor`*`]`
: Establecer la tonalidad predeterminada. *alteración* > 0 establece el número de sostenidos; *alteración* < 0 establece el número de bemoles. Se indica una tonalidad menor mediante `:1`.

`-o, --output=`*`archivo`*
: Escribir la salida en *archivo*.

`-s, --start-quant=`*`DURACIÓN`*
: Cuantizar el inicio de las notas a *DURACIÓN*.

`-t, --allow-tuplet=`*`DURACIÓN*NUMERADOR/DENOMINADOR`*
: Permitir duraciones de grupos especiales *DURACIÓN*NUMERADOR/DENOMINADOR*.

`-v, --verbose`
: Ser prolijo en comentarios.

`-V, --version`
: Imprimir el número de la versión.

`-w, --warranty`
: Presentar la garantía y el copyright.

`-x, --text-lyrics`
: Tratar todos los textos como letra de la canción.

Advertencias y problemas conocidos

Las notas superpuestas en un arpegio no se procesarán correctamente. La primera nota se lee y el resto se ignoran. Aplique la misma duración a todas las notas y añada marcas de fraseo o indicaciones de pedal.

5.2 Invocar `musicxml2ly`

MusicXML es un dialecto del XML para representar notación musical.

`musicxml2ly` extrae las notas, articulaciones, estructura de la partitura, letra, etc., de archivos de MusicXML parte a parte, y los escribe en un archivo .ly. Se invoca a través de la línea de órdenes.

La invocación desde la línea de órdenes se hace como sigue:

```
musicxml2ly [opción]... archivo_xml
```

Observe que por 'línea de órdenes', nos referimos a la línea de órdenes del sistema operativo. Consulte Capítulo 5 [Conversión desde otros formatos], página 28, para obtener más información acerca de esto.

Si el nombre de archivo proporcionado es '`-`', `musicxml2ly` lee la entrada desde la entrada estándar.

`musicxml2ly` contempla las siguientes opciones:

`-a, --absolute`
: convertir las alturas en modo absoluto.

`-h,--help`
: mostrar un resumen de la utilización y las opciones.

`-l, --language=IDIOMA`
: utilizar un archivo de idioma distinto 'IDIOMA.ly' y los correspondientes nombres de nota, p.ej. 'deutsch' para deutsch.ly y los nombres de nota en alemán.

`--lxml` usar el paquete de Python lxml.etree para el análisis de XML; usa menos memoria y tiempo de CPU.

`--nd --no-articulation-directions`
no convertir las direcciones (`^`, `_` o `-`) para las articulaciones, dinámica, etc.

`--no-beaming`
no convertir la información de las barras, en vez de ello usar el barrado automático de LilyPond.

`-o,--output=`*archivo*
fijar el nombre del archivo de salida como *archivo*. Si *archivo* es '`-`', la salida se imprime sobre stdout, la salida estándar. Si no se da, se usa *archivo_xml*'`.ly`'.

`-r,--relative`
convertir las alturas en modo relativo (predeterminado).

`-v,--verbose`
ser prolijo.

`-v,--version`
imprimir la información de la versión.

`-z,--compressed`
el archivo de entrada es un archivo MusicXML comprimido en zip.

5.3 Invocar `abc2ly`

ABC es un formato bastante simple basado en ASCII. Se encuentra descrito en el sitio web de ABC:

http://www.walshaw.plus.com/abc/learn.html.

`abc2ly` convierte ABC en LilyPond. Se invoca de la siguiente manera:

```
abc2ly [opción]... archivo_abc
```

`abc2ly` contempla las siguientes opciones:

`-b,--beams=None`
preservar la noción de ABC de las barras

`-h,--help`
esta ayuda

`-o,--output=`*archivo*
fijar el nombre del archivo de salida como *archivo*.

`-s,--strict`
ser estricto respecto al éxito

`--version`
imprimir la infomación de la versión.

Existe una posibilidad rudimentaria para añadir código de LilyPond el archivo fuente de ABC. Si decimos:

```
%%LY voices \set autoBeaming = ##f
```

Producirá que el texto que sigue a la palabra clave 'voices' se inserte en la voz en curso del archivo de salida de LilyPond.

De forma similar,

```
%%LY slyrics más palabras
```

producirá que el texto que sigue a la palabra clave 'slyrics' se inserte en la línea de letra en curso.

Advertencias y problemas conocidos

El estándar ABC no es muy 'estándar'. Existen diferentes convenciones para las posibilidades avanzadas (por ejemplo, polifonía).

No se pueden convertir varias melodías de un solo archivo.

ABC sincroniza las letras y las notas al principio de una línea; `abc2ly` no lo hace.

`abc2ly` ignora el barrado de ABC.

5.4 Invocar `etf2ly`

ETF (Enigma Transport Format) es un formato utilizado por Finale, un producto de Coda Music Technology. `etf2ly` convierte parte de un archivo ETF en un archivo de LilyPond listo para usar.

Se invoca a través de la línea de órdenes como sigue:

```
etf2ly [opción]... archivo_etf
```

Observe que por 'línea de órdenes', nos referimos a la línea de órdenes del sistema operativo. Consulte Capítulo 5 [Conversión desde otros formatos], página 28, para obtener más información acerca de esto.

`etf2ly` contempla las siguientes opciones:

`-h,--help`
: esta ayuda

`-o,--output=`*ARCHIVO*
: fijar el nombre del archivo de salida como *ARCHIVO*

`--version`
: información de la versión

Advertencias y problemas conocidos

La lista de inscripciones de articulación posibles es incompleta. Los compases vacíos confunden a `etf2ly`. Las secuencias de notas de adorno no se dan por finalizadas satisfactoriamente.

5.5 Generar archivos de LilyPond

El propio LilyPond no contempla la utilización de ningún otro formato, pero existen algunas herramientas externas que también generan archivos de LilyPond.

Entre ellas se encuentran las siguientes:

- Denemo, editor de partituras gráfico.
- Rumor, convertidor monofónico de MIDI a LilyPond en tiempo real.
- lyqi, un modo principal para Emacs.
- xml2ly, que importa MusicXML.
- NoteEdit que importa MusicXML.
- Rosegarden, que importa MIDI.
- FOMUS, una biblioteca de LISP para generar notación musical.
- `http://vsr.informatik.tu-chemnitz.de/staff/jan/nted/nted.xhtml`, contempla experimentalmente la exportación para LilyPond.
- `http://www.tuxguitar.com.ar/`, puede exportar a LilyPond.
- `http://musescore.org` también puede exportar a LilyPond.

Apéndice A 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 B Índice de LilyPond

\

\header dentro de documentos LaTeX 17

A

ABC 30
actualización de un archivo de LilyPond 11
advertencia 10
apuntar y pulsar 4
apuntar y pulsar, línea de órdenes 7
archivo de salida, tamaño del 4
archivos, búsqueda de 8

B

bugs (fallos) 13
búsqueda, ruta de 8

C

Coda Technology 31
colores, sintaxis 3
convert-ly 11

D

docbook 14
DocBook, música dentro de 14
documentos, insertar música en 14
dvips 24

E

editores 3
emacs 3
enigma 31
error 10
error de programación 10
error de Scheme 10
error fatal 10
error, formato de los mensajes de 11
error, mensajes de 10
ETF 31
`Extracción de fragmentos de música` 26

F

fallos (bugs) 13
fatal, error 10
FDL, GNU Free Documentation License 32
Finale 31
`First steps` 6

H

html 14
HTML, música dentro de 14

I

informes de fallo 13
invocación de dvips 24
invocar `lilypond` 6

L

LANG 10
latex 14
LaTeX, música dentro de 14
LILYPOND_DATADIR 10
llamadas, traza de 10

M

mensajes de error 10
MIDI 28
miniatura 19
modos del editor 3
musicología 14
MusicXML 29

O

opciones de la línea de órdenes para `lilypond` 6
OpenOffice.org 26
órdenes, opciones de la línea de 6

P

PostScript, salida 8
programación, error de 10
programas externos, generación de archivos de LilyPond 31

S

salida, establecer el formato de 8
Scheme, error de 10
Scheme, volcado de 8
sintaxis, resaltado de 3
`Staff` 28
SVG (Scalable Vector Graphics) 8

T

texi 14
texinfo 14
Texinfo, música dentro de 14
tipografías de outline 24
títulos en HTML 19
títulos y lilypond-book 17
traza de Scheme 10
type1, tipografías 24

V

version 11
versión de los archivos 11
vim 3
vista previa, imagen 19
`Voice` 28