

TEXemplares

Boletín de CervanTEX,
Grupo de Usuarios de TEX
Hispanohablantes

Año 5.º, número 6 (extraordinario), segundo trimestre 2004

TEXemplares

TEXemplares es el boletín de CervanTEX, el grupo de usuarios de TEX hispanohablantes. Queda prohibida cualquier reproducción total o parcial por cualquier medio, sea convencional o electrónico, de su contenido. Ni *TEXemplares* ni CervanTEX comparten necesariamente la opinión de los colaboradores. Nada en *TEXemplares* debe entenderse como una postura oficial del grupo.

Las colaboraciones deben ser creadas con la clase `TeXemplar.cls`, disponible por CTAN en una versión preliminar. Se debe usar, siempre que sea posible, caracteres de 7 bit y acentuar con el método de abreviaciones de babel. Las figuras externas en eps se deben crear a 600 pt.

Este ejemplar se creó con las siguientes aplicaciones: pdfTeX, Version 3.14159-1.10b, LaTeX2e <2001/06/01> y teTeX 2.0.2 junto con la clase `TeXemplar 2001/07/21 v0.4`, `Babel <v3.7h>` y `spanish.ldf 2001/01/30 v4.1c`. Se imprimió a 600 pt. con una ampliación de 1,414 y luego reducido a su tamaño real en las copias.

Notación. Por necesidades de composición y contrariamente al comportamiento normal de `\verb`, el código puede ser dividido a final de línea. Las divisiones en un símbolo no se indican mientras que aquellas entre letras se hace con un guión estilístico. Eso quiere decir que un guión a final de línea *nunca es parte del código* mientras que los que haya al comienzo *sí lo son*.

Redacción. Javier Bezos, Enrique Meléndez, Luis Seidel.

Depósito Legal y otros registros en trámite.

Editorial

Estimado lector,

Este número extraordinario de $\text{T}_{\text{E}}\text{X}$ emplares contiene la primera parte de las dos que componen las actas del tercer encuentro del Grupo de Usuarios de $\text{T}_{\text{E}}\text{X}$ Hispanohablantes, celebrado los días 15 y 26 de septiembre de 2003, en los locales del departamento de Matemáticas de la Universidad de Murcia.

En esta ocasión se presentan tres ponencias que tratan temas muy distintos entre sí. El primero describe el paquete `flexbib`, de José M. Mira, que simplifica la elaboración de la sección de referencias, contemplando un amplio conjunto de opciones para satisfacer casi cualquier formato. Francisco Alcaraz nos explica su experiencia en el uso de $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ en un dominio poco habitual: la botánica, usando el sistema operativo Linux. Finalmente, Salvador Sánchez-Pedreño hace una introducción a la edición de partituras musicales usando $\text{T}_{\text{E}}\text{X}$, centrándose en concreto en `MusiX $\text{T}_{\text{E}}\text{X}$` y sus utilidades anejas.

Este número ha sido posible gracias a la labor de los editores, Javier Bezos, Luis Seidel y Enrique Meléndez, quienes han dedicado mucho tiempo a que la calidad en su edición sea la adecuada.

El agradecimiento va también a los autores, que han aportado lo esencial del número, sus contribuciones. Se quiere animar desde aquí a todos los usuarios de $\text{T}_{\text{E}}\text{X}$ y amigos de $\text{T}_{\text{E}}\text{X}$ emplares a mantener vivo este boletín remitiendo contribuciones para próximos números.

Damos la palabra ahora a los organizadores.

Durante los días 25 y 26 de septiembre se celebró en Murcia, acogido por el Departamento de Matemáticas, el *Tercer Encuentro de Usuarios de T_EX Hispanohablantes* (EGUTH). Son varias las razones por las que podemos decir que este tercer encuentro tuvo un carácter especial.

De un lado es el primero que se realiza fuera de Madrid. Esta excentricidad geográfica es quizás una de las causas del marcado carácter local en la procedencia de los participantes, unida, desde luego, a la falta de compromiso con el encuentro por parte de los responsables y socios de CervanT_EX.

Por otro lado el programa del encuentro presentaba claras novedades respecto a los anteriores, fundamentalmente porque contenía una tendencia también excéntrica, en el más puro sentido de la palabra: salir del centro, del núcleo de T_EX-~~L~~A_TE_X, para ir hacia otros terrenos menos visitados, tanto en las materias (música, botánica, HTML y ~~L~~A_TE_X), como en el ámbito de uso (~~L~~A_TE_X en la imprenta profesional). Todo ello queda debidamente reflejado en el programa completo del encuentro, incluido tras esta presentación.

A pesar de los largos años pasados en el uso, enseñanza y difusión de T_EX y ~~L~~A_TE_X, su presencia fuera del ámbito académico sigue siendo rara, al menos en España. Como organizadores de este EGUTH estábamos, y estamos, convencidos de que es hora de cambiar esta tendencia, para lo que parecía oportuno, y del máximo interés, organizar una mesa redonda con profesionales de la imprenta y la tipografía. Mientras que los empresarios invitados a formar parte de la mesa respondieron favorablemente, no hicieron lo mismo los invitados a asistir a la discusión, resultando en una mesa redonda escasa en representación empresarial, bien es cierto que, en nuestro entorno geográfico, no abundan las empresas del ramo.

La mesa redonda resultó de gran interés tanto para los asistentes, como para los propios participantes. Aprovechamos este lugar para presentar algunas de las ideas puestas de manifiesto en ella, lugar adecuado, dado que no aparecerá referencia alguna a dicho acto en ninguna otra publicación. He aquí, pues, dichas ideas:

- La edición cuidada, la tipografía y composición de calidad, están en peligro.
- Con la excepción de muy contadas ediciones, la carestía de este trabajo está llevando a los profesionales a reclamar, cada vez más, que sea el usuario (autor, editor, etc.) el que proporcione un documento ya preparado, que el profesional imprimirá tal cual, dejando pues toda la responsabilidad al origen.

- Para la estrategia anterior el formato PDF es el adecuado (y, a su vez, puede estar facilitando esta tendencia).
- Incluso cuando en origen (autor, editor, etc.) se busca como intermediaria una empresa de edición, ésta a menudo se ve obligada a rebajar los índices de calidad de la composición tipográfica, a fin de ajustarse a bajos presupuestos. Con todo, algunas editoriales experimentan dificultades en la búsqueda de concedores del arte de la tipografía.
- T_EX y L^AT_EX pueden suponer una alternativa muy ventajosa en esta situación, ya que alcanzan altos niveles de calidad tipográfica, incluso en manos no expertas en esta disciplina. Desde esta perspectiva, el conocimiento profundo de L^AT_EX puede constituir, en el momento actual, un bagaje importante para jóvenes en busca de un empleo.
- Puede ser momento de empezar a considerar seriamente la conveniencia de incluir L^AT_EX en las programaciones curriculares de la formación profesional.

Resta únicamente dejar constancia de nuestro agradecimiento a todos aquellos que hicieron posible la realización del encuentro y que participaron en ella. En cuanto a los primeros, agradecemos el apoyo financiero de la Universidad de Murcia, la Fundación Séneca y la Comunidad Autónoma de la Región de Murcia. Respecto a los segundos vaya nuestro más cordial agradecimiento a los conferenciantes venidos del exterior: Piet van Oostrum y Santiago Muelas. Y finalmente gracias a los protagonistas en la mesa redonda: Joaquín Caravaca (de Pictografía), Isabel Capella (de Pearson-Prentice Hall), Jesús Sánchez (de Desktop Publishing), Francisco Torrano (de Formato-Formación y Secretario de la asociación de Artes Gráficas de Murcia) y Alejandro Pelegrín (de Compobell) que, aunque finalmente, por felices circunstancias, tuvo que ausentarse, fue el principal motor de la mesa redonda.

Es un placer haber contribuido a la confirmación de una estrecha relación de Murcia con L^AT_EX.

Los organizadores:
Bernardo Cascales Salinas
Pascual Lucas Saorín
José Manuel Mira Ros
Antonio Pallarés Ruiz
Salvador Sánchez-Pedreño Guillén

Tercer encuentro del grupo de usuarios de T_EX hispanohablantes

Programa

Información e inscripción en <http://www.latex.um.es/eguth03>.

Jueves 25 de septiembre

- 09:00 Recepción de participantes y recogida de documentación
- 10:00 Inauguración oficial
- 10:30 Piet van Oostrum (Institute of Information and Computing Sciences. Utrecht University): «Software engineering aspects of programming L^AT_EX (packages)»
- 11:30 Café
- 12:00 José Manuel Mira (Universidad de Murcia): «Gestión de bases de datos BibT_EX»
- 12:30 Antonio Pallarés (Universidad de Murcia): «Conversión a HTML con tex4ht»
- 13:00 Salvador Sánchez-Pedreño (Universidad de Murcia): «Edición de partituras musicales»
- 15:30 Reunión de grupos de trabajo de «CervanT_EX»
- 16:30 Café
- 17:00 Presentación de «El libro de L^AT_EX» (Prentice Hall, 2003) Autores: B. Cascales, P. Lucas, J. M. Mira, A. Pallarés y S. Sánchez-Pedreño
- 17.30 Mesa redonda: «L^AT_EX y la imprenta profesional»
- 21.00 Cena oficial

Viernes 26 de septiembre

- 09.30 Santiago Muelas (Universidad Politécnica de Madrid): «Metagraf, gráficos para L^AT_EX»
- 10.30 Francisco Alcaraz (Universidad de Murcia): «L^AT_EX, Linux y Botánica: una excelente combinación»

11.00 Café

11.30 Sesión de microaula

13.30 Clausura de las jornadas

Bibliografía flexible: el sistema flexbib

José M. Mira
Departamento de Matemáticas
Universidad de Murcia,
mira@um.es

.....

Resumen

La gestión mecanizada de la bibliografía con BIB_TE_X proporciona un importante nivel de comodidad para el usuario, con tal de que utilice un modelo de bibliografía de los contemplados en ciertos estilos estándar y, además, escriba en inglés. Pero el calificativo «estándar» realmente es un eufemismo porque la lista de estilos que se encuentran en CTAN resulta interminable... y es bastante fácil perderse en esa selva sin encontrar la respuesta buscada. Sorprendentemente, y a diferencia del nivel de estandarización y flexibilidad desarrollado para otros aspectos de L^AT_EX, la bibliografía no ha alcanzado el status que la hace accesibles a los usuarios noveles y cuestiones básicas, como la gestión de los idiomas, están sin automatizar.

En esta comunicación se hace una propuesta para mejorar la situación y se aportan algunos instrumentos para llevarla a cabo. Se utiliza un sistema que permite una gestión estandarizada y flexible, que incluye los idiomas y un amplio abanico de parámetros que simplifican la personalización de la bibliografía.

1. Introducción

Son muchas las tareas automáticas que L^AT_EX realiza, o puede realizar, para ayudar a los autores a preparar sus documentos: la bibliografía es una de ellas. Los elementos básicos de cualquier bibliografía son dos: la lista bibliográfica y las referencias a los ítems de dicha lista. En principio, una lista bibliográfica es únicamente una lista de ítems en la que van apareciendo ordenadamente los diferentes elementos de la bibliografía (libros, artículos en revistas, tesis,

proceedings, etc.) y para cada uno de ellos se puede incluir autor, título, año, revista y páginas... dando a estos elementos un formato que es conforme con determinadas pautas sobre utilización de versalitas, negritas, itálicas, comillas... para los autores, el año, el título, etc. Los diferentes ítems de la lista pueden aparecer identificados ya sea por un número, unas siglas o, simplemente, una sangría inversa. Obviamente, es posible construir listas bibliográficas y referencias de forma completamente manual, pero esa no es una elección acertada porque no impermeabiliza contra errores potenciales, no sólo en el caso de las listas numeradas, sino también en las otras.

La utilización del comando `\cite` encomienda a \LaTeX el control automático de la correspondencia entre las referencias y la lista bibliográfica, impidiendo así los errores antes aludidos. Ese es el primer nivel para la mecanización de la bibliografía con \LaTeX . Casi todos los usuarios de \LaTeX utilizan este comando. Pero existe un segundo nivel que, en mi opinión, es más útil aún: el empleo de $\text{BIB}\TeX$ y de bases de datos de bibliografía.

En nuestra dilatada experiencia en la enseñanza de \LaTeX hemos podido constatar que, sin embargo, son muchos los usuarios que no utilizan ese segundo nivel de mecanización. Además no existe correlación entre la utilización de $\text{BIB}\TeX$ y el «nivel» del usuario de \LaTeX : conocemos usuarios expertos de \LaTeX que no utilizan $\text{BIB}\TeX$ y usuarios noveles (alumnos nuestros) que están utilizando $\text{BIB}\TeX$ casi desde el comienzo de su aprendizaje y consideran que la gestión de la bibliografía a través de bases de datos es uno de los grandes atractivos del sistema \LaTeX . Por supuesto, que existen también usuarios avanzados que utilizan $\text{BIB}\TeX$.

En lo sucesivo me limitaré a la gestión automatizada de la bibliografía usando bases de datos bibliográficas para $\text{BIB}\TeX$, que es el marco en el se realiza esta comunicación. Actualmente incluso las propias bases de datos tienen una gestión confortable, porque además de existir interfaces gráficas para gestionarlas (mi preferida es JabRef, jabref.sourceforge.net) en muchas ocasiones el contenido de las mismas puede ser obtenido a través de internet (así ocurre en matemáticas: www.ams.org/mathscinet y www.emis.de/ZMATH).

En la sección siguiente aparece descrita la propuesta que hago, pero para poder comprenderla y valorarla es conveniente establecer el contexto en el que se formula y los trabajos anteriores sobre el tema: a estos aspectos están dedicados los párrafos que siguen.

El formato utilizado en las publicaciones científicas (libros, revistas, etc.) para elaborar la lista bibliográfica y el modo en que aparecen referenciados en el cuerpo del documento los ítems de dicha lista están lejos de ser homogéneos. Mientras que en algunas disciplinas es frecuente utilizar números entre

corchetes para referirse a los elementos de la lista bibliográfica, en otras, esa práctica resulta anecdótica y se utiliza en cambio el autor y el año con idéntico propósito. Y cada uno de estos hábitos admite variantes: el número puede estar rodeado de corchetes o paréntesis y su tamaño puede ser normal o de superíndice, y en el segundo la utilización de autor más el año entre paréntesis –por ejemplo, López (2000)–, ambos autor y año rodeados de un mismo paréntesis –como en (López, 2000)– son sólo un botón de muestra de la variedad de formatos utilizados para las referencias bibliográficas. Mayores aún son las diferencias entre las distintas formas de elaborar las listas bibliográficas, respecto a cursivas, negritas, versalitas, ubicación del año, etc. Ni siquiera dentro de la misma disciplina se siguen unos patrones uniformes. De hecho, son muchas las editoriales y revistas que tienen sus propias pautas.

Desde el punto de vista de la gestión automatizada de la bibliografía con $\text{BIB}\text{T}\text{E}\text{X}$ cada una de estas variantes requiere un estilo ad hoc de bibliografía (bst) para $\text{BIB}\text{T}\text{E}\text{X}$, y en el caso de las referencias con formato autor-año, se requiere, además, un paquete para $\text{L}\text{A}\text{T}\text{E}\text{X}$ que implementa comandos adicionales al comando `\cite` para establecer el formato de las citas en el texto. Tales paquetes han sido elaborados de forma independiente, y por tanto, aunque dos paquetes diferentes lleguen a obtener el mismo resultado (por lo menos en algunos aspectos), la sintaxis utilizada para conseguirlo suele ser distinta. Esto representa una complicación para los autores que se ven en la necesidad de utilizar diferentes estilos de bibliografía en sus escritos, dependiendo del destino de éstos.

El paquete `natbib` desarrollado por Daly [1] aporta un nivel importante de comodidad al implementar unos comandos de uso universal, para cierta clase de estilos de bibliografía, entre los que se encuentran los estilos estándar `plain` y `alpha`, junto a otros estilos de uso extendido como son `harvard`, `apalike`, `chicago`, `astron`, `authordate` y uno del propio Daly cuyo nombre es `natbib`. Usando este paquete los autores pueden obtener bibliografías conforme a los estilos antes aludidos sin necesidad de tener que ir cambiando los comandos de tipo `\cite` que aparezcan en su documento, les basta con ir cambiando el argumento del comando `\bibstyle`, pudiendo, de hecho, obtener formatos numéricos para la bibliografía a partir de estilos de bibliografía de tipo autor-año.

A pesar de los progresos que `natbib` aporta para flexibilizar la gestión de la bibliografía no hay que olvidar que se opera sobre un archivo `bb1` generado por $\text{BIB}\text{T}\text{E}\text{X}$ a partir de un estilo de bibliografía bst y que tales estilos son habitualmente muy cerrados. Temas tan básicos como la elección de los tipos de letra para autores o títulos no son modificables, ni siquiera está implementa-

do un funcionamiento «tipo babel» que, por ejemplo, sustituya la conjunción *and*, necesaria para separar en la base de datos los diferentes autores, por la conjunción que corresponda al idioma en uso. Ciertamente éstos son problemas ficticios si el trabajo va a ser publicado en una determinada editorial y se ha utilizado el estilo *bst* suministrado por la editorial; pero esa no es, si mucho menos, la única situación posible: el autor puede tener sus propios gustos o el editor tiene pautas concretas para la elaboración de la bibliografía a las que no dan respuesta la multitud de estilos *bst* disponibles, lo cual ocurrirá, con toda seguridad, si el idioma de la publicación no es el inglés.

Resulta llamativo que habiéndose realizado en \LaTeX estandarizaciones como las proporcionadas por los paquetes *babel*, *graphicx* o *color*, la situación sea tan precaria en cuanto a gestión estandarizada de la bibliografía. Hasta el punto de que, a los usuarios más veteranos, nos hace recordar antiguos calvarios por los que tuvimos que pasar, antes de que se elaboraran paquetes como los mencionados.

2. El sistema *flexbib*

La motivación de esta comunicación es proponer que es necesario desarrollar un sistema más flexible y confortable para gestionar la bibliografía y compartir el resultado del trabajo que he realizado, con esa finalidad, para atender a las demandas de ayuda que he recibido de algunos de nuestros alumnos, mi esposa en particular, y dar respuesta así a necesidades específicas de formatos bibliográficos no contemplados en los estilos estándar. He llamado *flexbib* a ese sistema y quienes estén interesados en él pueden encontrarlo en www.latex.um.es.

Para el desarrollo he aprovechado el trabajo realizado anteriormente por otras personas, especialmente por Daly, pero con una orientación complementaria, que persigue generar un fichero *bb1* lo suficientemente parametrizado como para permitir diferentes formatos de bibliografía a partir de un único fichero *bb1*, sin necesidad de tener que elaborar ficheros de estilo *bst* ni realizar nuevas compilaciones con \BIBTeX . Por razones de estandarización y economía de esfuerzos he procurado utilizar los comandos de Daly. En el momento actual las características más destacadas del sistema son:

1. En cuanto al idioma: permite realizar una elección al estilo *babel*, que tiene efectos sobre la traducción de determinadas partículas que aparecen en las listas bibliográficas (*and*, *and other*, *in*, *edition*...) o sobre hábitos relativos a la utilización de comillas en los diferentes idiomas.

2. En cuanto a las referencias:

- Se puede elegir un modelo numérico, un modelo autor-año (que admite distintas posibilidades) o un modelo mixto, autor-número, que consideramos preferible tanto al estilo `plain`, por la facilidad de lectura, como al estilo `alpha`, por razones de índole estético.
- Se puede elegir entre utilizar paréntesis o corchetes y, para el caso del modelo autor-año, controlar si los paréntesis rodean sólo al año, a ambos, autor y año o, simplemente, no existen.
- Cuando hay varias citas en el argumento de un comando `\cite` es posible elegir si aparecen o no ordenadas de acuerdo con el orden de la lista y, en caso de utilizar un modelo numérico, si los números se comprimen para producir en su lugar un intervalo numérico. También es posible, en este caso, decidir si entre las citas, como separador de las mismas, se utiliza la coma o el punto y coma.
- Cuando el número de autores es superior a dos, en el modelo de citas autor-año, puede elegirse si se utilizan todos los autores o sólo el primero seguido del grupo «y otros», o lo que corresponda según el idioma.
- En algunas disciplinas existe la costumbre de incluir como nota a pie de página los datos completos de una referencia bibliográfica cada vez que se utiliza por vez primera en un capítulo, usando posteriormente expresiones como «Opus citada» o «Ibidem». El paquete `flexbib` es compatible con esta costumbre.
- También es posible utilizar comandos adicionales a `\cite` para recuperar de forma independiente el autor o el año de un ítem de la bibliografía.
- Se puede elegir, de forma independiente, la fuente en que aparecerán el autor, la conjunción «y» entre autores y el grupo «y otros», o lo que corresponda según el idioma.

3. En cuanto a la lista bibliográfica:

- Se puede elegir, de forma independiente, el tipo de letra utilizado en los apellidos y nombres de autores o editores. También es posible seleccionar el tipo de letra para la conjunción que precede al último de los autores.

- La fecha se puede hacer aparecer a continuación de los autores o después de la información editorial.
- Cuando aparecen dos o más ítems en la lista con idénticos autores es posible elegir si se repiten en todos los ítems los autores o si, por el contrario, sólo aparecen en el primero de los ítems, siendo sustituidos en los restantes por una raya.
- Cuando hay varios autores, es posible elegir el separador que se utilizará entre ellos.
- Para el caso del título de los artículos es posible elegir el formato de éstos (tipos, eventuales comillas, etc.).
- También pueden controlarse los tipos para el año y el volumen.
- El campo estándar `note`, destinado a anotaciones intrínsecamente significativas para el elemento de la bibliografía, aparece reflejado siempre en la lista, pero existe además un campo `comment`, destinado a comentarios o anotaciones personales, cuya inclusión en la lista bibliográfica es optativa. El campo `abstract`, destinado a incluir un resumen del contenido, tiene un comportamiento similar a `comment`.
- Es posible seleccionar la separación entre los ítems así como la sangría inversa para el caso de las listas autor-año.

4. En cuanto a las bases de datos:

- A los campos contemplados en los estilos estándar de bibliografía (`plain`, `alpha`, `unsrt`, `abbrv` y `apalike`) creados por Oren Patashnik [3, 4] el sistema *flexbib* añade y gestiona otros nuevos.
 - `abstract` Destinado a incluir un resumen.
 - `comment` Destinado a anotaciones personales o comentarios.
 - `doi` Destinado a recoger el DOI (Digital Object Identifier) del elemento.
 - `eid` Destinado a recoger el EID (Electronic Identification) del elemento.
 - `hyphenation` Destinado a mejorar, en caso necesario, la división silábica en el título del elemento bibliográfico. El contenido de este campo es un idioma que será utilizado como argumento del comando `\selectlanguage` implementado en el paquete `babel` que, por tanto, debe estar cargado.

`isbn` Destinado a recoger el ISBN (International Standard Book Number) del libro.

`issn` Destinado a recoger el ISSN (International Standard Serial Number) del elemento.

`url` Destinado a recoger el URL (Universal Resource Locator) del elemento.

- A diferencia de otros estilos de bibliografía, `flexbib.bst` respeta las mayúsculas y minúsculas en todos los campos haciendo que el contenido de los mismos resulte más «limpio».
 - El contenido del campo `edition` sólo requiere el número, permitiendo así que la opción del idioma elegido complete la información adecuada.
5. El paquete `flexbib.sty` está destinado a trabajar en colaboración con el estilo de bibliografía `flexbib.bst` pero también puede usarse con otros estilos de bibliografía como: `plain`, `alpha`, `harvard`, `chicago`. En tal caso, `flexbib.sty` emulará el resultado que proporcionaría el paquete correspondiente, pero algunas de las opciones de `flexbib.sty` no estarán disponibles.

El correcto funcionamiento de las referencias cruzadas entre las citas y la lista bibliográfica requiere, como es habitual, dos compilaciones con \LaTeX después de haber generado con \BIBTeX el correspondiente fichero `bb1`.

3. Opciones del paquete

El paquete `flexbib` implementa varias opciones que son recogidas en la lista que sigue. Las que aparecen en un mismo ítem separadas entre sí por una raya vertical, son mutuamente excluyentes.

`spanish | english | french | german | ...` Traduce al idioma indicado determinados textos que aparecen en las citas y en la lista bibliográfica. La opción por defecto es `spanish`.

`authoryear | plain | plain* | super | footcite` Selecciona el estilo de las citas. En el primer caso se obtienen citas de tipo autor-año (opción por defecto). En el segundo se utiliza únicamente el número entre

corchetes (como en el estilo `plain`). El tercero es una variante del anterior que, además del número de la referencia, incluye los autores en formato abreviado. El cuarto utiliza únicamente el número, en formato superíndice. Y el último utiliza toda la información del correspondiente elemento de la bibliografía; en este caso el comando `\cite` se utiliza, típicamente, en el argumento de un comando `\footnote`. Existe también una opción `numbers`, por compatibilidad con `natbib`, que equivale a `plain`.

`round | square | curly | angle` Determina el formato de los «paréntesis» en las citas, correspondiendo dichas opciones, respectivamente, a paréntesis `()`, corchetes `[]`, llaves `{}` y ángulos `<>`.

`colon | comma` Determina el separador a utilizar entre las citas cuando en el argumento de un comando `\cite` hay varias citas. La opción `colon` separa las citas con punto y coma y es el comportamiento por defecto para las opciones `authoryear` y `super`. La opción `comma` separa las citas con coma y es el comportamiento por defecto para la opción `plain`.

`sort` Cuando hay varias citas en el argumento de un comando `\cite` éstas aparecen ordenadas, si la opción ha sido seleccionada, de acuerdo con el orden de la lista bibliográfica. Si la opción no ha sido seleccionada, el orden de las citas respeta el utilizado en el argumento del comando `\cite`.

`sort&compress` Con las opciones `plain` y `super`, ordena las citas en el argumento de un mismo comando `\cite` y las agrupa en intervalos, si es posible. Esta opción y la anterior proporcionan las prestaciones de los paquetes `citesort` y `cite`, que no deben ser cargados.

`longnamesfirst` Cuando una obra tiene tres o más autores, el comando `\cite` únicamente incluye el apellido del primero de ellos seguido del valor almacenado en el comando `\bbbletal`, que, en el caso de utilizar la opción `spanish`, corresponde a «y otros». Con esta opción se consigue que la primera vez que se utilice un comando `\cite` para esa obra se incluyan los apellidos de todos los autores.

`nonamereplace` Cuando un autor (o grupo de autores) aparece repetido en la lista bibliográfica, `flexbib` lo sustituye por una raya; esta opción desactiva tal sustitución.

`datebegin` | `dateend` Permite controlar la posición de la fecha en los ítems de la lista bibliográfica. En los estilos de cita autor-año es frecuente que el inicio de cada ítem en la lista bibliográfica reproduzca el formato de la cita en el texto: la opción `datebegin` genera ese comportamiento. La opción `dateend` (opción por defecto) hace aparecer la fecha a continuación de la «información editorial», lo cual es también un formato frecuente para listas bibliográficas.

Cuando en un registro los campos `author`, `editor` y `key` están ausentes, con la opción `datebegin` el correspondiente ítem de la lista bibliográfica se inicia con el año, siendo el resultado obtenido menos satisfactorio en términos estéticos que el que se obtiene con la opción `dateend`. Hay que señalar asimismo que ambas opciones, en el caso de los campos antes mencionados estén vacíos, utilizan los tres primeros caracteres de la etiqueta identificativa del registro para la cita en los modelos autor-año.

`noabstract` Cuando un registro contiene un campo `abstract` el contenido de éste, si no es vacío, aparece en la lista bibliográfica. Con esta opción se suprime la inclusión en la lista del campo `abstract`.

`nocomment` Análoga a la opción anterior, pero referida al campo `comment`.

`openbib` Tiene el sentido ordinario: se inicia una nueva línea con cada bloque determinado por un comando `\newblock`.

`sectionbib` La lista aparece incluida en una unidad `\chapter*`, para la clase `book` y en una `\section*` para la clase `article`. Con esta opción se consigue que en ambos casos la lista bibliográfica se incluya en una unidad `\section*`.

`nonamebreak` Incluye en las citas los diferentes autores de una obra dentro de un comando `\mbox` evitando así la separación de éstos. Como consecuencia, pueden producirse líneas largas o cortas.

Conviene tener en cuenta que algunas de las opciones seleccionan otras. Así, por ejemplo, la opción `authoryear` selecciona por sí misma las opciones `round` y `colon`; mientras que `plain` selecciona `square`, `comma` y `dateend`. Esa selección por defecto puede ser cambiada, colocando la opción adecuada después de éstas.

4. Sobre el formato de las citas en el texto principal

Cuando Oren Patashnik creó su programa BiB \TeX y diseñó los cuatro estilos estándar de bibliografía (*plain*, *alpha*, *unsrt* y *abbrv*) únicamente consideró formatos numéricos para las citas. Cuando más tarde (¿a petición de usuarios de su programa?) diseñó el estilo *apalike*, con formato autor-año, dejó contundentemente reflejado, en el preámbulo de su nuevo estilo, su oposición a esa forma de realizar las referencias bibliográficas. Abrir la posibilidad de referencias no numéricas (práctica utilizada en muchas disciplinas científicas) conlleva la diversificación del comando `\cite`, abriendo así un abanico de posibilidades inexistentes en los estilos numéricos. Esta sección se ocupa de los aspectos relacionados con esa diversidad.

Extensiones del comando `\cite`

Además del comando `\cite` usual, el paquete `flexbib.sty` incluye otros comandos que ofrecen posibilidades adicionales para las citas.

```

\citet[Anterior] [Posterior] {Etiqueta}
\citep[Anterior] [Posterior] {Etiqueta}
\citealt[Anterior] [Posterior] {Etiqueta}
\citealp[Anterior] [Posterior] {Etiqueta}

\citet*[Anterior] [Posterior] {Etiqueta}
\citep*[Anterior] [Posterior] {Etiqueta}
\citealt*[Anterior] [Posterior] {Etiqueta}
\citealp*[Anterior] [Posterior] {Etiqueta}

```

La referencia a la cita bibliográfica insertada en el texto principal por estos comandos incluye autor y año, en varios formatos.

- Usando `\citet` se consigue que el año vaya entre paréntesis, mientras que en `\citep` se consigue que los paréntesis rodeen al grupo entero: autor más año. En ambos casos se utiliza la forma abreviada para listar los autores, lo que significa que cuando son tres o más, únicamente aparece el primero de ellos siendo los demás sustituidos por el genérico «y otros» (o lo que corresponda según el idioma). En las versiones con asterisco se listan todos los autores.
- Las versiones `citeal...` suprimen los paréntesis y las diferencias entre ellas se ilustran en el ejemplo 1.

- Pueden utilizarse uno o dos argumentos optativos. Si se utiliza sólo uno se identifica con el segundo de ellos y es equivalente al único argumento optativo del comando `\cite` estándar, es decir, va en posición *Posterior*. El argumento *Anterior* aparece antes de la etiqueta básica y su utilización requiere cumplimentar, aunque sea con contenido vacío, el segundo argumento optativo.

El argumento *Etiqueta*, como es habitual, hace referencia al identificador del registro en la base de datos.

El comando `\cite` equivale, dependiendo de las opciones, a alguno de los comandos anteriores. Obviamente es posible cambiar ese comportamiento sin más que reasignar el comando `\cite` al comando de la lista anterior que se desee, por ejemplo, mediante un comando `\let`.

EJEMPLO 1

| | |
|--|---|
| <code>\citet{johnson90,per91}</code> | JOHNSON y otros (1990); PÉREZ y otros (1991) |
| <code>\citep{johnson90,per91}</code> | (JOHNSON y otros, 1990; PÉREZ y otros 1991) |
| <code>\citep{johnson90,johnson91}</code> | (JOHNSON y otros, 1990, 1991) |
| <code>\citep{luc90a,luc90b}</code> | (LUCAS y otros, 1990a,b) |
| <code>\citet[específicamente][pág. 57]{johnson90}</code> | JOHNSON y otros (específicamente 1990, pág. 57) |
| <code>\citep[específicamente][pág. 57]{johnson90}</code> | (específicamente JOHNSON y otros 1990, pág. 57) |
| <code>\citealt{johnson90}</code> | JOHNSON y otros 1990 |
| <code>\citealp{johnson90}</code> | JOHNSON y otros, 1990 |
| <code>\citealt[pg.~32]{jon90}</code> | JOHNSON y otros 1990, pg. 32 |
| <code>\citealp[pg.~32]{jon90}</code> | JOHNSON y otros, 1990, pg. 32 |
| <code>\citealp*{johnson90}</code> | JOHNSON, MARTÍNEZ y PAULI, 1990 |
| <code>\citealp{johnson90,per91}</code> | JOHNSON y otros, 1990; PÉREZ y otros, 1991 |

Diferentes formatos para el conjunto autor-año

El autor y el año pueden también ser recuperados de forma independiente con ayuda de los comandos que siguen.

| | |
|------------------------------------|-------------------------------------|
| <code>\citeauthor{Etiqueta}</code> | <code>\citeauthor*{Etiqueta}</code> |
| <code>\citeyear{Etiqueta}</code> | <code>\citeyearpar{Etiqueta}</code> |

Los comandos `\citeauthor` y `\citeyear` sólo incluyen, respectivamente, el autor y el año. Ambos sin paréntesis. El comando `\citeauthor*` es como `\citeauthor` salvo que incluye la versión larga de los nombres. Mientras que `\citeyearpar` inserta el año entre paréntesis.

EJEMPLO 2

| | | |
|---------------------------------------|-------------------|---------------------------|
| <code>\citeauthor{johnson90}</code> | <code>\par</code> | JOHNSON y otros |
| <code>\citeauthor**{johnson90}</code> | <code>\par</code> | JOHNSON, MARTÍNEZ y PAULI |
| <code>\citeyear{johnson90}</code> | <code>\par</code> | 1990 |
| <code>\citeyearpar{johnson90}</code> | | (1990) |

Diferentes formatos para el autor o el año

Los comandos anteriores tienen la flexibilidad suficiente para poder realizar cómodamente citas bibliográficas acordes con los hábitos de buena parte de las disciplinas científicas, que suelen utilizar referencias con formatos autor-año o numérico. Pero en algunas disciplinas existe la costumbre de incluir, la primera vez que una obra es citada en una unidad de estructura (habitualmente un capítulo), generalmente como una nota a pie de página, los datos completos de la obra. A tal fin se implementa el comando

```
\citefull{Etiqueta}
```

que incorpora la cita en el texto con el mismo formato utilizado para esa referencia en la lista bibliográfica general, salvo que no incorpora los campos `abstract` y `comment`. Habitualmente, dicho comando, formará parte del argumento de un comando `\footnote`. Además suprime el punto final con el que acaban todos los ítems de la bibliografía. La puntuación se introduce manualmente en el texto principal haciendo posible producir resultados como los siguientes.

EJEMPLO 3

| | |
|---|---|
| ...en tal sentido~\citefull{Mar00}. | ... véase en tal sentido MARTÍNEZ, ANTONIO: <i>Evolución y genética de poblaciones</i> . Universidad de Murcia, 2000. |
| ...en tal sentido~\citefull{Mar00}, p.~206. | ... véase en tal sentido MARTÍNEZ, ANTONIO: <i>Evolución y genética de poblaciones</i> . Universidad de Murcia, 2000, p. 206. |

Modificación de los tipos y de la puntuación en las citas

Para controlar el formato de los tipos empleados en las referencias bibliográficas se utilizan los siguientes comandos de un argumento.

```
\citenamfont    \citeandfont    \citetalfont    \citenumfont
```

El argumento coincide con los autores (o editores) en el caso del comando `\citenamfont`; la conjunción que liga el último autor con los anteriores, para el caso de `\citeandfont`; el grupo «y otros» (o lo que corresponda según el

idioma), para el caso de `\citetalfont`; y los números (en el estilo numérico) que aparecen en el texto para referirse a los elementos de la lista bibliográfica en el caso de `\citenumfont`. Los valores por defecto son asignados, en ese orden, por `flexbib.sty`, el fichero de configuración local `flexbib.cfg`, si existe, y las redefiniciones que hayan sido realizadas en el documento utilizando `\renewcommand`.

EJEMPLO 4

```
... la versión por defecto~\cite{Mar00}
\renewcommand{\citenamefont}[1]{#1}
... la versión modificada~\cite{Mar00}
```

```
... la versión por defecto MARTÍNEZ (2000)
... la versión modificada Martínez (2000)
```

Modificación de los tipos en las citas

Como ya se ha señalado, las opciones al cargar el paquete `flexbib` tienen efectos sobre el formato de los paréntesis y signos de puntuación utilizados en las citas. Pero también es posible fijar estos elementos mediante el siguiente comando con seis argumentos obligatorios.

```
\bibpunct{#1}{#2}{#3}{#4}{#5}{#6}
```

| Arg. | Valor def. | Función |
|------|------------|---|
| #1 | (| Paréntesis de apertura. |
| #2 |) | Paréntesis de cierre. |
| #3 | ; | Puntuación entre citas múltiples. |
| #4 | | Estilo de los números en citas «numéricas»: n normal, s superíndice |
| #5 | , | Puntuación entre autor y año en modo <code>\citep...</code> |
| #6 | , | Puntuación entre años (o números) para una cita cuando el autor común desaparece. Si autor y año son idénticos aparecen en la forma '1994a,b'; si se desea añadir un espacio hay que hacerlo en el argumento #6 en la forma { , ~}. |

5. Sobre el formato de las listas bibliográficas

Si las referencias numéricas admiten gran variedad de formatos, las posibilidades para las listas bibliográficas son aún mayores. De ahí, en mi opinión, la conveniencia de construir ficheros `bb1` suficientemente parametrizados para dar cabida a esas posibilidades, no como un mero ejercicio formal, sino

para permitir que los usuarios menos avanzados puedan obtener fácilmente listas bibliográficas automatizadas acordes con las hábitos más generalizados. En esta sección se describen comandos encaminados a conseguir esa finalidad.

En este primer grupo se incluyen los comandos y declaraciones relativos a los tipos utilizados para determinados elementos de la lista bibliográfica y para el conjunto de la misma (el último de ellos).

| | | |
|----------------------------|-------------------------------|-------------------------|
| <code>\bibnamefont</code> | <code>\bibyearfont</code> | <code>\enquote</code> |
| <code>\bibfnamefont</code> | <code>\bibvolfont</code> | <code>\bibnumfmt</code> |
| <code>\bibandfont</code> | <code>\bibinfont</code> | <code>\bibfont</code> |
| | <code>\inbooktitlefont</code> | |

Los de la primera columna controlan, respectivamente, el formato de los tipos en los apellidos de autores y editores, en el nombre de pila de autores y editores y en la conjunción (o el separador) que precede al último de los autores, cuando son varios.

Los de la segunda columna controlan, respectivamente, el formato de los tipos en el año, en el volumen y número de las revistas, en la partícula «En» (o lo que corresponda según el idioma) que se incluye cuando un trabajo ha aparecido publicado «En» un libro, actas de congreso, etc. cuyo título se indica y, finalmente, en el título del libro en los registros `inproceedings` o `incollection`.

Finalmente los de la tercera columna controlan, respectivamente, el formato de tipos en el título de los artículos (en la configuración por defecto se utiliza también para incluir comillas); el formato de los tipos utilizados para los números en una lista bibliográfica numerada; y los parámetros comunes a toda la lista.

Salvo el último que es una declaración (comando sin argumento) los restantes son comandos de un argumento y pueden ser redefinidos, además de en `flexbib.cfg`, en el propio documento utilizando `\renewcommand` de forma análoga a la ejemplificación realizada en el ejemplo 4 a propósito del comando `\citenamfont`. Para modificar la declaración `\bibfont` se procede de forma similar, pero sin utilizar la variable #1. Así, por ejemplo mediante

```
\renewcommand{\bibfont}{\small}
```

se obtendría una lista bibliográfica con unos tipos de tamaño inferior a los utilizados en el resto del documento.

Los comandos del siguiente grupo son declaraciones destinadas a controlar elementos de separación o sustitución de autores.

| | | |
|-----------------------------|----------------------|---------------------------|
| <code>\nameseparator</code> | <code>\bbland</code> | <code>\namereplace</code> |
|-----------------------------|----------------------|---------------------------|

El primero de los comandos declara el separador entre los autores o editores (salvo para el último de ellos) cuando son más de dos; su valor por defecto es punto y coma. El segundo realiza la misma función de separación entre el último autor y el que le precede; su valor por defecto es la conjunción «y» (o lo que corresponda de acuerdo con el idioma).

Cuando un autor (o grupo de autores) aparece repetido en la lista es una práctica frecuente sustituirlo por un filete. El comando de un argumento (el autor) `\namereplace` puede ser redefinido para que el argumento variable sea reemplazado por una constante (la raya); por ejemplo, mediante

```
\renewcommand{\namereplace}[1]{---{ }---
```

se conseguiría la raya que corresponde a la yuxtaposición de dos guiones largos.

La declaración

| |
|---------------------------|
| <code>\bibpreamble</code> |
|---------------------------|

puede ser utilizada para incluir comentarios, o cualquier material precediendo a la lista bibliográfica. Inicialmente está definida como vacía, pero utilizando `\renewcommand` puede ser redefinida como el contenido de tales comentarios. A diferencia de lo que ocurre con la lista propiamente dicha, el contenido de `\bibpreamble` utiliza los tipos normales del documento (salvo que explícitamente se fijen otros en su redefinición) y no se ve afectado por la declaración `\bibfont` que determina el formato global de los tipos en la lista bibliográfica.

Las longitudes

| | |
|----------------------|-----------------------|
| <code>\bibsep</code> | <code>\bibhang</code> |
|----------------------|-----------------------|

controlan, respectivamente, la separación vertical entre los ítems de la lista bibliográfica y la sangría inversa en las listas tipo autor-año.

Los campos `abstract` y `comment`

Salvo que se utilice la opción `noabstract` (`nocomment`), el contenido del campo `abstract` (`comment`) aparece en la lista después de los demás campos y claramente diferenciado de ellos.

Cuando un registro contiene un campo `abstract`, al final del correspondiente `\bibitem` del archivo `bb1` se incluye el comando

```
\bibabstract{Etiqueta}{ContenidoDelCampoAbstract}
```

El *ContenidoDelCampoAbstract* es arbitrario. Si contuviera muchos caracteres (más de 5000, para la Version 0.99c) podría llegar a saturar la capacidad de gestión de BIB_TE_X, ignorándose en tal caso dicho registro. Para resolver el problema sería necesario guardar dicho contenido en un fichero externo a la base de datos e incluir en el campo *abstract* un comando `\input` cuyo argumento fuera el nombre de dicho fichero.

Por defecto, el comando `\bibabstract` incorpora a la lista bibliográfica el *ContenidoDelCampoAbstract* dentro de un entorno *quotation* precedido de un antetítulo y no utiliza el argumento *Etiqueta*. El valor de dicho antetítulo coincide con el guardado en la declaración `\abstractname` del paquete *babel*.

El campo *comment* se comporta de forma análoga a *abstract* y existe, correspondientemente, un comando

```
\bibcomment{Etiqueta}{ContenidoDelCampoComment}
```

y un antetítulo, dependiente del idioma, y que se guarda en la declaración `\bibcommentname`.

Redefiniendo `\bibabstract` o `\bibcomment` mediante `\renewcommand` es posible controlar el formato utilizado para incluir los referidos campos.

Idiomas

No conozco una herramienta para la gestión automatizada de la bibliografía con BIB_TE_X cuya funcionalidad sea similar a la que proporciona el paquete *babel*: la traducción de determinadas palabras o frases de acuerdo con el idioma elegido, sin necesidad de introducir cambios en el cuerpo del documento. El sistema *flexbib* emplea esa misma técnica de utilizar comandos cuyas asignaciones son realizadas en una serie de ficheros auxiliares, como: *spanishbst.tex* (español), *englishbst.tex* (inglés), *frenchbst.tex* (francés) y *germanbst.tex* (alemán) por ahora, aunque utilizándolos como modelo pueden construirse también ficheros análogos para otros idiomas.

Comentarios finales

Cuando se escribe con L_AT_EX un trabajo cuya producción final la realiza totalmente el autor del mismo, una gestión automatizada de la bibliografía como la propuesta en esta comunicación puede resultar muy atractiva debido a su confortabilidad. Pero no siempre es así y resulta natural preguntarse ¿qué ocurre si ese trabajo ha de ser enviado a un editor de una revista para que lo

ensamble con otros? Esa es exactamente la situación que se nos ha presentado en esta comunicación.

BIB \TeX genera la lista bibliográfica almacenándola en un fichero, cuyo nombre coincide con el del documento principal y cuya extensión es `bb1`, utilizando para ello los comandos `\bibliography` y `\bibliographystyle`. Tal fichero, que no es otra cosa que un entorno `thebibliography`, es el que \LaTeX usa en la siguientes compilaciones para construir la bibliografía. Por consiguiente, nada impide sustituir los comandos antes citados por el contenido del correspondiente fichero `bb1`, ya que, con independencia del procedimiento usado para construir el entorno `thebibliography`, ahí está toda la información necesaria para que \LaTeX genere la lista bibliográfica y las referencias a ella en el texto principal: base de datos y estilo de bibliografía resultan ya elementos completamente innecesarios.

En el caso de los estilos estándar de bibliografía, que únicamente emplean comandos «universalmente» comprensibles para \LaTeX , eso es todo lo que hay que hacer, pero en caso de utilizar en el entorno `thebibliography` comandos no «universalmente» comprensibles para \LaTeX (*flexbib* trabaja de ese modo para conseguir la parametrización buscada), dichos comandos han de ser definidos mediante un paquete para \LaTeX o bien en el propio entorno. El sistema *flexbib* es consciente de ello y genera un fichero `bb1` suficientemente autónomo para conseguir buena parte de las prestaciones del sistema (en el formato `plain`), sin necesidad de que éste esté presente. Ello permite además que el autor o el editor puedan, si lo desean, hacer retoques posteriores en el contenido del fichero `bb1` de forma muy sencilla y con carácter general para todos los ítems de la bibliografía.

6. Ilustraciones

Los ejemplos que siguen muestran algunas de las posibilidades del sistema *flexbib*.

EJEMPLO 5

```
\usepackage{flexbib}

\begin{document}

\section*{Las citas en el texto}
...\cite{chomsky-73} ... \cite{croft-tesis}
...\cite{croft-80}

\bibliography{test}
\bibliographystyle{flexbib}
```

Sin opciones (spanish y authoryear están defecto)

Las citas en el texto

...CHOMSKY y otros (1973) ...CROFT (1978) ...CROFT (1980)

Referencias

CHOMSKY, N. y LÓPEZ, J. M.: «Condiciones de transformación». **En:** S. R. Anderson y P. Kiparsky (Eds.), *A festschrift for Morris Halle*, Holt, Rinehart & Winston, New York, 1973.

CROFT, W. B.: *Organizing and searching large files of document descriptions*. Tesis doctoral, Cambridge University, 1978.

— : *New models on organizing document descriptions*. *Journal of European Models*, 1980, **69**, pp. 55-89.

EJEMPLO 6

```
\usepackage[plain]{flexbib}

\begin{document}

\section*{Las citas en el texto}
...\cite{chomsky-73} ... \cite{croft-tesis}

\bibliography{test}
\bibliographystyle{flexbib}
```

Opción plain

Las citas en el texto

...[2] ...[1]

Referencias

[1] CHOMSKY, N. y LÓPEZ, J. M.: «Condiciones de transformación». **En:** S. R. Anderson y P. Kiparsky (Eds.), *A festschrift for Morris Halle*, Holt, Rinehart & Winston, New York, 1973.

[2] CROFT, W. B.: *Organizing and searching large files of document descriptions*. Tesis doctoral, Cambridge University, 1978.

EJEMPLO 7

```
\usepackage[plain,french]{flexbib}

\begin{document}

\section*{Las citas en el texto}
...\cite{chomsky-73} ... \cite{croft-tesis}

\bibliography{test}
\bibliographystyle{flexbib}
```

Opciones plain y french (obsérvese el espacio en las comillas)

Las citas en el texto

...[2] ...[1]

Referencias

[1] CHOMSKY, N. et LÓPEZ, J. M.: « Condiciones de transformación ». **Dans:** S. R. Anderson y P. Kiparsky (Réds.), *A festschrift for Morris Halle*, Holt, Rinehart & Winston, New York, 1973.

[2] CROFT, W. B.: *Organizing and searching large files of document descriptions*. Thèse de doctort, Cambridge University, 1978.

EJEMPLO 8

```
\usepackage[datebegin]{flexbib}

\begin{document}

\section*{Las citas en el texto}
... \cite{chomsky-73} ... \cite{croft-tesis}
... \cite{croft-80}

\bibliography{test}
\bibliographystyle{flexbib}
```

Opción datebegin

Las citas en el texto

...CHOMSKY y otros (1973) ...CROFT (1978) ...CROFT (1980)

Referencias

CHOMSKY, N. y LÓPEZ, J. M. (1973). «Condiciones de transformación». **En:** S. R. Anderson y P. Kiparsky (Eds.), *A festschrift for Morris Halle*, Holt, Rinehart & Winston, New York.

CROFT, W. B. (1978). *Organizing and searching large files of document descriptions*. Tesis doctoral, Cambridge University.

— (1980). *New models on organizing document descriptions*. *Journal of European Models*, **69**, pp. 55-89.

EJEMPLO 9

```
\usepackage[datebegin,nonamereplace]{flexbib}
\renewcommand{\bibnamefont}[1]{#1}
\renewcommand{\bibfnamefont}[1]{#1}
\renewcommand{\citenamefont}[1]{#1}

\begin{document}

\section*{Las citas en el texto}
... \cite{croft-tesis} ... \cite{croft-80}

\bibliography{test}
\bibliographystyle{flexbib}
```

Opciones datebegin, nonamereplace y cambios en los tipos

Las citas en el texto

...Chomsky y otros (1973) ...Croft (1978) ...Croft (1980)

Referencias

Croft, W. B. (1978). *Organizing and searching large files of document descriptions*. Tesis doctoral, Cambridge University.

Croft, W. B. (1980). *New models on organizing document descriptions*. *Journal of European Models*, **69**, pp. 55-89.

Referencias

- [1] DALY, PATRICK W.: «Natural Sciences Citations and References», 2003. Documentación que acompaña al paquete natbib.
- [2] DOWNES, MICHAEL: «The amsref package», 2002. Documentación que acompaña al paquete.
- [3] PATASHNIK, OREN: «BIB_TE_Xing», 1988a. Documentación que acompaña al programa BIB_TE_X.
- [4] —: «Designing BIB_TE_X styles», 1988b. Documentación que acompaña al programa BIB_TE_X.

L_AT_EX, Linux y la Botánica: una excelente combinación

Francisco J. Alcaraz Ariza
Departamento de Biología Vegetal
Universidad de Murcia
e-mail: falcaraz@um.es

.....

1. Introducción

En la Botánica, como en otras disciplinas científicas, la concepción de colectividad suele venir acompañada de unas actitudes de compartimento estanco, que dificultan el intercambio, no sólo de opiniones, sino de métodos e ideas con otros campos de la ciencia. Esta forma de actuar es la más extendida y eso a pesar de que muchos campos de la investigación sobre los vegetales precisan de la colaboración con otras disciplinas científicas:

- **Arqueología:** reconocimiento de restos vegetales en yacimientos arqueológicos.
- **Bioquímica:** sustancias químicas de los vegetales útiles para el hombre o que ayudan a la diferenciación de especies y a la interpretación filogenética de las mismas.
- **Climatología:** relaciones de los vegetales con el clima.
- **Ecología:** relaciones de los vegetales con el medio y con otros organismos.
- **Edafología:** relaciones entre el suelo y la cubierta vegetal.
- **Fisiología vegetal:** aspectos de funcionamiento de los vegetales.

- Genética: selección y mejora de vegetales de interés para el hombre, relaciones filogenéticas entre los vegetales.
- Geografía: relaciones de las plantas con el hombre y con la geomorfología.
- Geología: relaciones de los vegetales con las rocas que afloran en la superficie terrestre.
- Matemáticas: análisis numérico de datos sobre taxonomía o ecología de los vegetales y las comunidades de organismos en las que participan.
- Medicina: efectos de las plantas sobre la salud humana, alergias producidas por el polen y otros elementos de las plantas.
- Nutrición: aspectos nutricionales de los vegetales.
- Paleontología: estudio de los fósiles de vegetales.
- etc.

Pero estas colaboraciones no suelen trascender más allá de la pura investigación; incluso cuando como resultado de las mismas se producen publicaciones científicas, lo habitual es que si por sus contenidos son enviadas a revistas propias de la otra disciplina, sean los especialistas de la misma que han colaborado en el proyecto conjunto los que se encarguen de preparar los manuscritos.

En este orden de cosas, si los otros grupos que han participado en el desarrollo de la investigación conjunta utilizan otras aplicaciones informáticas y otro modo de organizar y presentar las publicaciones, la influencia sobre los botánicos participantes es prácticamente nula.

De forma más concreta, nuestra experiencia en el ámbito de las publicaciones científicas, comunicaciones a congresos, informes, redacción de tesis doctorales y cualquier otro tipo de documentos que resultan de la investigación botánica, pone de relieve que fundamentalmente se utiliza un procesador de textos de un sistema propietario, el más extendido en el mundo de la informática personal hoy en día, hasta el punto de que son las propias revistas científicas o los contratistas de los proyectos de investigación los que exigen el formato, también propietario, de tal procesador de textos en todo tipo de documentos que le sean enviados.

En otras palabras, el binomio estándar de las publicaciones en el ámbito de la botánica es el tan conocido de un Sistema Operativo de la serie Windows

(98, 2000, XP) y el procesador de textos MsWord, con los correspondientes archivos en formato «doc».

Un repaso a las páginas en castellano sobre Botánica nos revela esa monolítica realidad, en la que el uso de sistemas operativos diferentes de los que tienen su sede en la lluviosa ciudad de Seattle y el uso de procesadores de textos distintos del MsWord, y no digamos nada de alguna aplicación que no sea Wysiwyg¹, es un acto de pura fantasía. Hasta tal punto esto es así que en el ámbito de la Botánica no se habla de aplicaciones en modo genérico; Windows es el sistema por excelencia y en vez de decir «procesador de textos» se habla de *Word* o frente al término «presentaciones» se dice, simple y llanamente, *Power Point*.

El entorno de las publicaciones y redacción de documentos en el ámbito de la docencia e investigación de la Botánica española es, por tanto, muy hostil para la introducción de alternativas, por muy superiores en prestaciones y resultados que éstas sean a las que se usan mayoritariamente.

En este sentido, podemos decir, de forma aproximada pero sin temor a desviarnos demasiado de la realidad, que entre los investigadores botánicos españoles más del 70 % desconoce la existencia de sistemas operativos alternativos a los de Windows (Freebsd, Linux, Solaris, etc.) y prácticamente el 100 % no sabe lo que representa el término L^AT_EX.

Es ciertamente un panorama desalentador, pero que esperamos empiece a resquebrajarse en un futuro próximo; creemos que hay diversos indicios en el entorno que van a darle un mayor protagonismo al software libre en el ámbito de la Botánica, entre ellos:

- El constante aumento de los precios de las licencias y actualizaciones de los sistemas operativos Windows.
- Las amenazas que sobre la seguridad de los datos siguen siendo las notables deficiencias de dichos sistemas operativos, con grandes agujeros de seguridad frente a todo tipo de ataques, de los que los relativos a «virus informáticos» son los más conocidos, aunque no los únicos ni necesariamente los más dañinos.
- El control que suponen las actuales formas de activar los programas propietarios y que se va a profundizar enormemente con la aplicación de la tecnología *Palladium*.

¹Abreviatura de «What you see is what you get»; es decir, lo que se ve es lo que se tiene, refiriéndose que se edita el documento con el aspecto en pantalla que se obtendrá al imprimirlo

- La inestabilidad del sistema, que supone la pérdida de muchas horas de trabajo.
- La competencia con opciones libres al MsWord, entre las que destaca particularmente OpenOffice.org, por su carácter multiplataforma, la gran fiabilidad en la producción, si es necesaria, de ficheros en formato «doc» y sobre todo la capacidad de exportación a pdf y el uso de un formato abierto de archivos (XML) que además produce ficheros de tamaño más reducido.
- La capacidad de \LaTeX para producir documentos de gran calidad en los que la ubicación de las figuras y cuadros, las citas bibliográficas, las referencias cruzadas y la creación de índices y tablas de contenido es de una potencia inimaginable en los procesadores de textos «Wysiwyg» al uso.

En las siguientes secciones del presente artículo se va a exponer nuestra experiencia con el uso de Linux y \LaTeX en el campo de la docencia e investigación científica en Botánica, como muestra de que es posible usar vías alternativas y de que éstas, que se hacen muy cuesta arriba al principio, conllevan la obtención de rendimientos mucho mayores.

La mayor parte de las soluciones que se han ido aquilatando en el desarrollo de textos botánicos bajo \LaTeX me han sido proporcionadas por otros usuarios más avanzados de este entorno o las he encontrado en páginas de Internet y la bibliografía utilizada; en particular quiero expresar aquí el reconocimiento a todas las personas de la lista de correo de \LaTeX en español (sptex@ls.cica.es), ya que unas veces directamente, respondiendo a requerimientos míos, y otras de forma indirecta, al aportar soluciones a problemas planteados por otros usuarios, han solucionado todas mis dudas hasta el momento.

2. La problemática

A lo largo de nuestra experiencia en el campo de la Botánica, hemos podido observar como hay deficiencias en los documentos producidos como resultado de la docencia e investigación en este ámbito científico, las cuales tienen un carácter calificable de endémico y son además muy difíciles de subsanar con los procesadores de texto al uso; entre otras:

- Estructuración muy deficiente de los textos (capítulos, secciones, subsecciones), con abuso de subapartados, lo que dificulta la comprensión de

los documentos y la ubicación del lector en las diferentes partes de los mismos.

- Se suelen usar demasiados «tipos de fuentes» y tamaños de las mismas.
- El cambio de una fuente por otra no suele seguir las normas que dicta la ortodoxia en el ámbito de la edición; tampoco se presta demasiada atención a la legibilidad de los caracteres en los diversos apartados del texto.
- Están al día las inconsistencias en las citas y referencias bibliográficas:
 - Falta de homogeneidad en la forma de presentar las citas en el texto.
 - Citaciones en el texto que no se recogen en la bibliografía.
 - Referencias bibliográficas que no aparecen citadas en el texto del documento.
 - Erratas en las referencias bibliográficas.
- Escaso y deficiente uso de las referencias cruzadas.
- Configuración muy deficiente de los índices generales (tablas de contenido, cuadros y figuras), siendo común que no se presente nada más que el de contenidos.
- Confusiones frecuentes entre cuadros y tablas o entre figuras y mapas, así como en la forma de numerarlos.
- Formatos poco apropiados de cuadros, especialmente en el caso de los de vegetación, que frecuentemente deben ocupar varias páginas.
- Escaso uso de los índices alfabéticos temáticos.
- No se resuelve el problema derivado de la inestabilidad de los nombres científicos, lo que produce en documentos cuya preparación temporal es larga (tesis, informes) inconsistencias importantes a lo largo del texto.

3. Soluciones bajo Linux

Se podría decir que estamos rizando el rizo cuando no sólo apoyamos el uso de L^AT_EX como alternativa a MsWord, sino que además proponemos la migración desde los sistemas operativos de Microsoft hacia Linux; sin embargo,

tras tres años usando, al principio de forma testimonial, diversas distribuciones de este sistema operativo de código abierto, apreciamos tanto las ventajas frente a los inconvenientes, que la propuesta de cambio está sobradamente justificada.

Entre los aspectos más relevantes que estimamos hacen del uso de Linux una alternativa viable y, en muchos sentidos, muy conveniente a los sistemas de Microsoft, queremos destacar:

- Se puede obtener de forma gratuita y está permitida la instalación en tantas computadoras como sea necesario².
- Hay distribuciones que permiten tener en funcionamiento equipos que bajo sistemas operativos propietarios serían obsoletos, al no alcanzar los requerimientos mínimos que aquellos precisan.
- También es posible reciclar muchos de esos equipos antiguos usándolos como terminales que trabajan en un servidor, papel que desempeña un ordenador más potente. Se prolonga así notablemente la vida útil de las computadoras.
- Linux es un sistema operativo muy robusto y estable por su esquema de configuración surgido de Unix, esto se traduce en ahorros sustanciales del tiempo empleado en el mantenimiento de los equipos.
- Básicamente un Linux bien configurado e instalado en una computadora en la que no haya problemas de hardware ni se usen programas en pruebas, no se cuelga. Los cuelgues en Linux son debidos a algún tipo de problema específico, nunca aleatorio, por lo que es posible investigar y seguirle la pista al problema, detectarlo y, por tanto, corregirlo.
- Si Linux se usa siguiendo ciertas normas básicas la posibilidad de verse afectado por virus, gusanos y troyanos es nula. Esto supone un ahorro enorme en tiempo dedicado a corregir las secuelas de tales ataques.
- Linux se actualiza «en caliente»; se evitan así los reinicios tras la actualización de las aplicaciones.
- A diferencia de lo que ocurre con los sistemas Windows, las actualizaciones en Linux suponen siempre una mejora notable del código, por lo que

²La idea de gratuidad es relativa, pues para las personas implicadas en el mundo del código abierto suele ser habitual el apoyo económico al desarrollo de las distribuciones que utilizan, si bien esta es una opción, no una obligación.

el sistema en sí y las aplicaciones mejoran su rendimiento y disminuyen los tiempos de espera; es decir, a cada nueva distribución la computadora funciona mejor.

- Muchas de las distribuciones de Linux más recientes (Fedora, Mandrake, Suse, etc.) se instalan fácilmente y configuran de forma automática la computadora y sus periféricos. De hecho son cada vez más los fabricantes que ya incluyen «controladores» de sus periféricos para Linux. Además, ya empiezan a venderse equipos con Linux preinstalado, una de las grandes bazas por las que el uso de Windows está tan generalizado.
- Tras la instalación de una distribución Linux está disponible en la computadora la mayor parte de las aplicaciones que se van a necesitar; esto supone un ahorro considerable en el tiempo de puesta en marcha de las mismas.

Como consecuencia, usar Linux supone un considerable ahorro para los grupos de investigación en tiempo y en gastos para la adquisición de equipos informáticos y software, además de contar con una notable prolongación de la vida útil de los equipos.

Uno de los tópicos que existe sobre Linux es el de la falta de aplicaciones; sin embargo es evidente que esto no es así, lo que se puede comprobar simplemente echando un vistazo a la siguiente página de Internet:

<http://linuxshop.ru/linuxbegin/win-lin-soft-en/table.shtml>

En general, la programación en Linux sigue la línea filosófica surgida de Unix, consistente en realizar aplicaciones de pequeño tamaño y que realicen tareas muy concretas de forma inmejorable, que programas de gran tamaño que realicen muchas cosas a la vez. Cuando se necesita realizar varias tareas a la vez se puede crear una aplicación gráfica desde la que se pueda «llamar» a los distintos programas menores.

Esta forma de programar ha dado como resultado aplicaciones muy probadas, por lo que están optimizadas para la tarea específica que cada una lleva a cabo. Además, existen lenguajes que facilitan mucho la creación de los interfaces gráficos desde los que se lanzan las diversas aplicaciones.

Un ejemplo de este modo de proceder lo tenemos en un programa bajo Linux para facilitar el uso de L^AT_EX, se trata de *Xtexsh*, que es una aplicación gráfica desde la que se lanzan los diversos programas necesarios para crear un fichero de texto, editarlo, compilarlo y crear los diversos archivos de salida

(véase figura 1). Xtextsh es altamante configurable, de modo que el usuario puede elegir qué programa hace cada tarea y pasarle parámetros (L^AT_EX, editor de textos, creación de ficheros dvi, ps, pdf, programas para visualizar dichos ficheros, programa de revisión ortográfica, etc.).

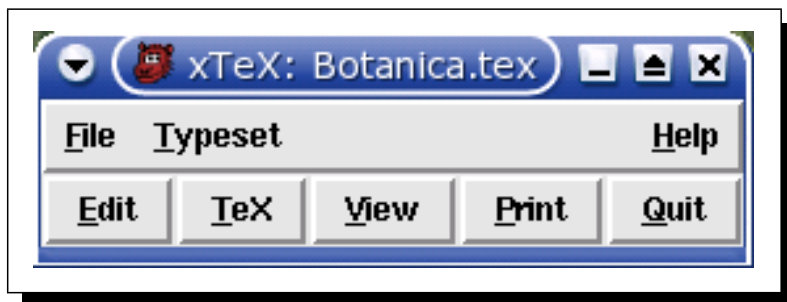


Figura 1. Aspecto de Xtextsh

Otra posibilidad, menos usual en Linux, es la de integrar el código de al menos algunos de los programas en una única aplicación; sigue esta filosofía, entre otros, el programa *Kile*, que recuerda por su aspecto actual (versión 1.6 en el momento de escribir el presente artículo) al conocido «Winedt» de Windows. En este caso no se ha integrado el código de todas las aplicaciones posibles, de modo que aún es posible seleccionar algunas para tareas concretas (visores de dvi, ps y pdf; corrector ortográfico, etc.).

En definitiva, el uso de L^AT_EX bajo Linux no presenta ningún problema e incluso está facilitado por el hecho de que todas las distribuciones modernas de Linux incluyen entre los paquetes básicos los necesarios para trabajar con L^AT_EX (frecuentemente se trata de T_ET_EX).

4. Paquetes y comandos útiles

Las peculiaridades de la publicación y creación de documentos en el campo de la Botánica nos ha llevado a ir adoptando diversas soluciones, alguna de las cuales puede diferir notablemente de las seleccionadas en otras disciplinas científicas; en otros casos se trata de soluciones específicas para determinados problemas o situaciones que se dan en los textos de Botánica. En los párrafos siguientes vamos a comentar sucintamente algunos de los aspectos que han sido de mayor relevancia al respecto.

Las citas y las referencias bibliográficas

La forma de citar publicaciones en un texto y de configurar las referencias bibliográficas en el ámbito de la Botánica se ajusta bastante bien a la configuración del paquete *chicago*. Este paquete permite una gran cantidad de opciones en la forma de presentar las citas en el texto:

- Autor (año)
- (Autor, año)
- (Autor, año1, año2)
- (Autor1, año; Autor2, año)
- etc.

Para que el lector pueda hacerse una idea mejor de cómo se presentan las referencias bibliográficas en las revistas al uso en Botánica, los siguientes items presentan respectivamente un ejemplo de la referencia de un libro, de un capítulo de libro y de un artículo en una revista científica botánica; el paquete *chicago* permite obtener esta estructura en un documento L^AT_EX sin ningún tipo de problema.

Webster, J. 1980. *Introduction to fungi*. Cambridge University Press. Cambridge.

Whatley, J.M. y Whatley, F.R.1984. Evolutionary aspects of the eukaryotic cell and its organells. *In*: Linskens, H.F. y Heslop-Harrison, J. (eds.), *Cellular interactions*: 345-368. Encyclopedia of plant physiology, New Series. Springer Verlag. Berlín.

Wof, P.G. 1997. Evaluation of *atpB* nucleotide sequences for phylogenetic studies of ferns and other Pteridophytes. *Amer. J. Bot.* 84: 1429-1440.

Cuadros, tablas, figuras, mapas y láminas

La distinción entre cuadros (con datos numéricos, no numéricos o de ambos tipos, pero dispuestos en tal forma que no son susceptibles de operaciones aritméticas) y tablas (con datos numéricos que son susceptibles de utilizarse en operaciones matemáticas, como por ejemplo sumas para dar totales) es clara, pero sin embargo ambos objetos son confundidos con frecuencia en las publicaciones botánicas. Por defecto al usar el paquete *babel* estos objetos flotantes se denominan «cuadros»; sin embargo es relativamente fácil crear

un nuevo tipo de objetos tipo «float» utilizando el paquete del mismo nombre, con el nombre, en este caso, de «tabla»:³:

```
\usepackage{float}
.....
\newfloat{tabla}{hbtpt}{cdr}
\floatname{tabla}{Tabla}
```

Cada tabla se crea en el texto poniendo:

```
\begin{tabla}
\caption{Mi tabla}\label{cdr:mi_tabla}
\end{tabla}
```

Finalmente, es posible poner una lista (índice) de tablas incluyendo en el lugar apropiado del documento:

```
\listof{tabla}{Lista de Tablas}
```

De modo similar se puede proceder en la distinción entre figuras, mapas e imágenes, algo también de gran utilidad en los documentos botánicos.

Los cuadros de vegetación

El estudio de la cubierta vegetal se lleva a cabo a través de diversos tipos de muestreo, seleccionados según las características de la vegetación a estudiar y las características del problema que se pretende resolver con el mismo.

El resultado de esta labor son cuadros voluminosos cuyo núcleo lo componen varias columnas, una de ellas, normalmente la situada más a la izquierda, en la que se presentan los nombres científicos de las especies identificadas y varias columnas más, tantas como muestras de vegetación se hayan levantado en el campo, en las que aparecen datos numéricos que suelen hacer referencia a parámetros como la cobertura (porcentaje de suelo cubierto por la proyección ortogonal del total de individuos de una especie observado en la parcela de muestro), la densidad, el número de individuos, etc.).

En territorios de flora pobre, como sucede en los países del norte de Europa, estas tablas no son muy largas, pues suelen ser pocas las especies vegetales identificadas en el conjunto de las parcelas; sin embargo en España la situación es diametralmente opuesta, no en balde hablamos del territorio con

³Este truco me lo proporcionó en la lista de L^AT_EX en castellano «Enrique», cuyo apellido desconozco; sirva esta nota para expresar mi agradecimiento.

mayor biodiversidad vegetal de la Comunidad Europea. Es así que alguno de estos cuadros puede tener más de 100 filas, correspondientes a otras tantas especies de plantas, por lo que es imposible encajarlas en una sola página. El paquete *longtable* nos ha permitido realizar cuadros de vegetación de una gran calidad, muy por encima de la que permiten las limitadas posibilidades que tienen los procesadores de texto al uso (MsWord, OpenOffice.org, etc.).

Referencias cruzadas

La expresión de las referencias cruzadas es mucho más sencilla y elegante si se usa el paquete *varioref*, el cual permite que de forma automática en el texto compilado no se indique la página donde está ubicado el elemento al que se hace referencia si se ubica en la misma desde la que se hace la llamada, figure el término «página anterior» o «página siguiente» en el caso de su ubicación en esa posición respecto a la llamada, o figure el número de página si su ubicación respect a la llamada no es ninguna de las tres mencionadas.

Basta con cargar al principio del documento el paquete con la opción «spanish»:

```
\usepackage [spanish] {varioref}
```

Y después hacer la llamada a la etiqueta usando el comando:

```
\vref{<<etiqueta>>}
```

Sin embargo vimos que el paquete en su opción para el castellano tenía algún error que originaba algunos resultados indeseados, como era el número de página pegado al texto «en la página». Pese a nuestro desconocimiento en L^AT_EX el problema era muy sencillo, el texto que se adjunta debe sustituir a la declaración para español que aparece en el fichero «varioref.sty»:

```
\DeclareOption{spanish}
  {\vref@addto\extrasspanish{%
    \def\reftextfaceafter {en la p\'agina siguiente}%
    \def\reftextfacebefore{en la p\'agina anterior}%
    \let\reftextafter      \reftextfaceafter
    \let\reftextbefore     \reftextfacebefore
    \def\reftextcurrent   {en esta p\'agina}%
    \def\reftextfaraway#1{en la p\'agina \pageref{#1}}%
    \def\reftextpagerange#1#2{en las p\'aginas
      \pageref{#1}--\pageref{#2}}%
  }}
```

```
\def\reftextlabelrange#1#2{\ref{#1} to~\ref{#2}}%
}}
```

El resultado con este cambio es prácticamente perfecto.

Los nombres científicos de los vegetales

Los nombres científicos de los vegetales están regulados por varios códigos internacionales de nomenclatura⁴. Aunque estos códigos imponen criterios para asegurar la estabilidad de los nombres científicos, han sido tantos los avances de la Botánica en las últimas décadas que no son infrecuentes los casos en los que la aplicación del código correspondiente impone la sustitución de un nombre científico ampliamente utilizado por otro que resulta «prioritario». Cuando los documentos son grandes y se van escribiendo a lo largo de varios años (por ejemplo, el texto de una tesis de doctorado) no es raro que el nombre científico de una planta que, de acuerdo con el código de nomenclatura botánica, ha cambiado durante el periodo de desarrollo del documento, aparezca de dos formas diferentes.

Ante este problema optamos por una solución que nos permite evitar ese efecto indeseado, consistente en, sólo para aquellos nombres científicos que se sabe están sometidos a procesos de análisis nomenclatural, por lo que cabría la posibilidad de que cambiaran en poco tiempo, hacer definiciones al principio del documento de los mismos, de modo que en el texto usaremos un comando para hacer referencia al nombre científico de la susodicha planta, pero éste sólo aparece realmente escrito en la definición correspondiente.

Por ejemplo, el nombre científico de la «carrasca», árbol típico del bosque mediterráneo en la Península Ibérica, tiene una cierta inestabilidad, oscilando según los autores entre los dos siguientes:

1. *Quercus rotundifolia*
2. *Quercus ilex* subespecie *ballota*

La decisión final sobre cuál de los dos utilizar no tenemos porque tomarla al inicio de escribir la tesis, sino que podemos esperar hasta el último momento; para ello podemos definir un comando (carrasca) en el preámbulo del documento del modo siguiente:

```
\def\carrasca{\emph{Quercus rotundifolia}}
```

⁴Téngase en cuenta que los vegetales incluyen organismos de varios reinos: *Bacteria*, *Protozoa*, *Chromista*, *Fungi* y *Plantæ*, algunos de ellos se rigen por códigos de nomenclatura diferentes.

En este caso hemos decidido que el nombre sea el primero de los comentarios y además, siguiendo la ortodoxia botánica para los nombres en latín, que se escriba destacado del resto texto. Si al final del trabajo se opta por otra opción, bastará cambiar el resultado del comando, sin tener que repasar todo el texto.

Cada vez que queramos escribir el nombre científico de la carrasca en el texto simplemente pondremos el comando y obtendremos el resultado apetecido⁵:

Quercus rotundifolia \`carrasca`

La escritura de los nombres científicos es siempre complicada y se cometen muchos errores, sobretodo si la persona encargada de mecanografiar el texto no es especialista. Por eso el uso de este tipo de definiciones puede ser de una gran utilidad.

Por otra parte, incluso los nombres científicos lo suficientemente estables pueden ser complicados de teclear, pues deben estar destacados del texto, típicamente dentro de un entorno de texto en letra redonda deberían aparecer en cursiva; además deben estar en latín, lo que añade posibilidades de errores al teclearlos por ser algo extraños para nuestro idioma. Ante esta problemática, en el documento que realizamos como informe que recogía los resultados finales de un trabajo sobre la flora y vegetación de humedales en la provincia de Murcia, para las familias de plantas, que aparecían repetidas veces a lo largo del texto, se utilizaron definiciones como la que se presenta a continuación:

`\def\Alismataceas{\emph{Alismatace\ae}}`

De este modo para citar a una familia en el texto usábamos el comando correspondiente al nombre castellanizado de la misma y el resultado fue espectacular, pues se obtenía el nombre en latín, con el símbolo especial «æ» al final del mismo, y el tipo destacado del resto del texto:

Alismataceæ \`Alismataceas`

⁵Mi agradecimiento a Gustavo Bustamante por enseñarme la confección del entorno example, con el que he podido presentar estos ejemplos en los que se enfrenta un comando L^AT_EX a su resultado en el documento final.

5. A modo de conclusión

Se da fin así a esta breve exposición de la experiencia de un botánico en un mundo doblemente hostil dentro de esta disciplina en España:

- El del uso de Linux como sistema operativo.
- El del manejo de \LaTeX para la creación de documentos.

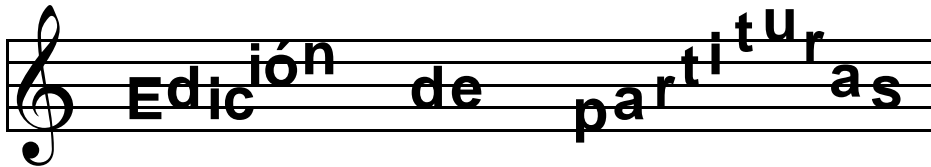
Tras la travesía del desierto que supusieron los inicios en ambos campos, podemos concluir que la experiencia no sólo ha sido positiva, sino que en estos, sus primeros pasos, ha mostrado unas posibilidades enormes para facilitar la creación de documentos sobre aspectos docentes y de investigación en Botánica.

El resultado ha sido la confección de las primeras tesis de doctorado con \LaTeX en botánica, diversos informes finales de proyectos, apuntes para alumnos (documentos pdf puestos en Internet), plantillas de documentos de uso habitual en la Universidad (convocatorias de exámenes, exámenes, documentos oficiales, solicitudes diversas), así como formularios para colecciones de comandos y «trucos» de gran utilidad.

Aunque nuestros colegas prácticamente son desconocedores del tema, es nuestra intención dar conocimiento de la existencia de todas estas posibilidades y en este sentido ya hemos presentado una comunicación a un congreso de Botánica sobre algunas de sus posibilidades⁶.

Evidentemente la inercia hace que haya una profunda resistencia a probar algo nuevo, con la idea de que si lo que se usa previamente funciona para qué se va a cambiar; pero, como se justificaba al principio del presente artículo, los resultados en la creación de documentos en Botánica dejan mucho que desear, por lo que más tarde o más temprano deberán abrirse a las inmensas posibilidades que brindan tanto Linux como \LaTeX .

⁶XIX Jornadas de Fitosociología, La Laguna, septiembre de 2003: Alcaraz, F. *et al.* *Optimización de los recursos informáticos en Geobotánica.*



Salvador Sánchez-Pedreño Guillén
Departamento de Matemáticas
Universidad de Murcia
e-mail:pedreno@um.es

.....

Resumen

Este texto representa una breve introducción a la edición de partituras musicales en $\text{T}_{\text{E}}\text{X}$ y su entorno. Se centra en MusiX $\text{T}_{\text{E}}\text{X}$, aunque se hacen algunas ligeras referencias a paquetes, precompiladores o entornos gráficos más o menos ligados a MusiX $\text{T}_{\text{E}}\text{X}$.

1. Motivación

Hacía ya mucho tiempo que deseaba poder intervenir en un encuentro EGUTH para presentar algunas ideas en torno a la edición de partituras musicales con $\text{T}_{\text{E}}\text{X}$, especialmente con MusiX $\text{T}_{\text{E}}\text{X}$. La ocasión la pintaban calva. Aún con la falta de tiempo que suponía participar en la organización del tercer EGUTH me decidí a hacerlo en mi propia tierra.

Un querido compañero del departamento me comentaba personalmente, tras mi intervención, que había quedado enormemente contento con ella; poco más o menos éstas fueron sus palabras: «*Me ha parecido una charla estupenda puesto que has hablado de algo estéticamente atractivo y que, además, no interesaba a nadie más que a ti mismo*». Por ello, convertir la charla en un texto escrito supone una motivación adicional: ¿conseguiré que mi buen amigo deje de tener razón?

En todo caso tanto la versión hablada como ésta escrita estuvieron desde el principio condenadas al fracaso y ello por varias razones. Ninguna de ellas podía tener la longitud suficiente para enseñar a los que no saben MusiX \TeX , ni tampoco podía descubrir nada a los que ya lo conocen. Ninguna de ellas interesaría a los usuarios de \TeX que no sean músicos, ya que prácticamente no entraría en las «profundidades» técnicas de MusiX \TeX , pero tampoco interesaría a los músicos que no conocen \TeX , porque seguramente ya poseen un sistema *mucho más simple y eficiente* para la edición de partituras.

En el mercado existen varios sistemas WYSIWYG para la edición de partituras musicales; entre los más potentes se encuentran: FINALE, SIBELIUS, GRAPHIRE MUSIC PRESS, los tres comerciales. Los músicos con experiencia en estas o similares herramientas seguramente se sienten a gusto con ellas, de ahí que probablemente las califiquen como *más simples y eficientes* y lo que está en juego, al menos para mí, es sembrar una duda acerca de MusiX \TeX y compañía: ¿podrá este sistema tan complicado, opuesto a la filosofía WYSIWYG, desplazar, aunque sólo sea parcialmente, a dichos sistemas?

En mi opinión «la compañía MusiX \TeX » es enormemente ventajosa en la edición de obras musicológicas. En éstas la longitud de los fragmentos musicales es pequeña y tenemos a nuestra disposición toda la potencia y calidad de \TeX y \LaTeX . Componer (tipográficamente hablando) con MusiX \TeX una obra musical completa (una sinfonía, una colección de obras para piano, etc.) puede ser una tarea tan compleja que resultaría difícil argumentar en su favor, si bien la capacidad de aprendizaje, la experiencia y algunas herramientas adicionales podrían hacerla rentable.

2. Un poco de historia — Pequeño homenaje a Daniel Taupin

A continuación presento un rápido recorrido por la evolución de los sistemas de edición de partituras ligados a \TeX . No es una historia sencilla.

- La idea de utilizar \TeX para la composición tipográfica de partituras musicales se remonta a algún momento entre 1986 y 1987, cuando aparece $M^{\mu}\TeX$ ¹, creado por Andrea Steinbach y Angelika Schofer, y que constituía (parte de) una Master's Thesis de la Rheinische Friedrich-Wilhelms University de Bonn. Este proyecto mereció la atención de François Jalbert, que, en 1989, escribió un manual en inglés [4] y propuso numerosas correcciones al trabajo original. El propio Jalbert parece dar testimonio de

¹Probablemente denominado inicialmente `mtex.tex`, un conjunto de macros para \TeX .

un cierto misterio en torno a las autoras Steinbach y Schofer: imposible contactar con ellas, desaparecieron sin huella. $\text{M}^{\text{U}}\text{T}_{\text{E}}\text{X}$ puede crear partituras sobre un único pentagrama, aunque puede manipular dos o más voces y conoce la mayor parte de los símbolos de acentuación.

- En 1992 Daniel Taupin creó $\text{MusicT}_{\text{E}}\text{X}$. Partiendo de las fuentes diseñadas en METAFONT por las anteriores autoras, aunque con un 100 % de correcciones, como el propio Taupin afirma.
- $\text{MusicT}_{\text{E}}\text{X}$ representó un hito pero mantenía serios inconvenientes en la calidad final, fundamentalmente en lo relacionado con las ligaduras de fraseo (para los no músicos: curvas sobre las notas que se extienden a lo largo de varios compases y, muy a menudo, a lo largo de varias líneas o páginas). Estas limitaciones llevan a Daniel Taupin, con la colaboración de Ross Mitchell y Andreas Egler, a crear $\text{MusixT}_{\text{E}}\text{X}$, que nació en 1995 (4695 líneas de código, versión actual T.112, 3 enero 2003).
- Posteriormente la historia se complica: Andreas Egler, por *razones personales*, se separa de Daniel Taupin (una cierta e intangible sombra de acritud se proyecta sobre este episodio) y continúa trabajando en su propia versión de $\text{MusixT}_{\text{E}}\text{X}$. La versión de Egler, en la que trabaja en colaboración con Stanislav Kneifl y Youping Huang, pasa a denominarse $\text{OpusT}_{\text{E}}\text{X}$ [6]. Actualmente la pista de $\text{OpusT}_{\text{E}}\text{X}$ se difumina: aparentemente continúa en versiones β y la última data aproximadamente del año 2001. Parece que el uso más intensivo de $\text{OpusT}_{\text{E}}\text{X}$ se centra en la escritura de música gregoriana.
- Los tipos METAFONT de $\text{MusixT}_{\text{E}}\text{X}$ tienen una restricción fundamental: no pueden ser generados a más de 600 ppp, pues la capacidad de METAFONT se ve excedida. En palabras del propio Taupin [8]:

I know this problem and there is NO solution: it is an overflow by metafont [...]. In other words, MusixT_EX is limited to 600dpi and the font characters are TOO big for metafont in 1200dpi and more.

Changing this is a tough job which requires complete revision of font management within musixtex.tex itself...

Felizmente en julio de 2001 Takanori Uchiyama distribuye las mismas fuentes en formato PostScript, creadas a partir de las fuentes originales de METAFONT, con ayuda de la utilidad $\text{T}_{\text{E}}\text{XTRACE}$ de Peter Szabo, con algunos toques manuales.

- Se crean varias herramientas que simplifican el lenguaje y la escritura de partituras, basados en MusiX \TeX : son los llamados precompiladores, es decir, sistemas con lenguaje más sencillo que, una vez procesado el documento fuente, proporcionan un documento en el lenguaje de MusiX \TeX . Entre ellos destacan: M \mathbb{P} , que ya no es mantenido, y que ha derivado en LILYPOND, una herramienta de enorme potencia (véase la sección 9), PMX y M-TX. Existe también un interface gráfico para Linux, denominado NOTEEDIT, que exporta a MusiX \TeX .
- MusiX \TeX viene acompañado de una gran cantidad de paquetes adicionales con objetivos muy diversos (tablaturas para guitarra, música gregoriana, ampliación de las capacidades básicas, etc.), pero además se crean otros paquetes mas generales:
 - musixser y musixlyr, ambos de Rainer Dunker [2, 3]. El primero permite codificar música unidimensionalmente, es decir, por voces que, posteriormente, se reunen automáticamente para proporcionar un sistema polifónico, posiblemente con varios pentagramas. El segundo aporta facilidades para la introducción de los textos en música vocal.
 - De gran interés son musixps (también denominado slursK) y slursM, paquetes que insertan ligaduras directamente en código PostScript, obteniendo resultados muy superiores a las ligaduras construidas mediante tipos. El primero, creado por Stanislav Kneifl a partir de su trabajo en Op μ S \TeX , está diseñado para DVIPS y similares, el segundo, de Hiroaki Morimoto, utiliza código METAPOST, accesible a DVIPDFM.

El futuro de MusiX \TeX es desconocido. Sin duda el inesperado y triste fallecimiento de Daniel Taupin, el 26 de agosto de 2003, es decir, exactamente un mes antes de la celebración del encuentro EGUTH, es la causa fundamental de esta incertidumbre, pues hasta ese momento Daniel Taupin se mantenía perfectamente activo con MusiX \TeX , publicando actualizaciones, con las correspondientes mejoras, con envidiable frecuencia.

Taupin murió en un accidente, mientras escalaba en los Alpes. Su pasión por la montaña comenzó a la edad de once años; en Francia era bien conocido como autor de varios importantes libros sobre escalada y por defender prácticas respetuosas con la montaña en esta disciplina deportiva. En el mundo de \TeX , además de crear MusiX \TeX naturalmente, participó en la creación de LTX2RTF, un conversor de \LaTeX a formato rtf, y en la conversión de tipos TrueType a METAFONT.



Daniel Taupin, 1936–26 de agosto de 2003

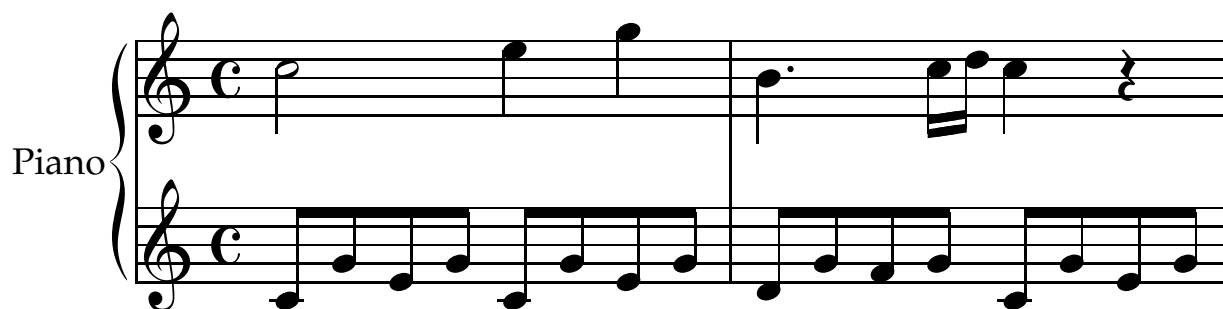
MusiX $\text{T}_{\text{E}}\text{X}$ no sólo es una materialización de la genial idea de adaptar $\text{T}_{\text{E}}\text{X}$ a la edición de partituras, resultando en una herramienta de gran belleza y potencia, sino que viene acompañado de un magnífico manual: *MusiX $\text{T}_{\text{E}}\text{X}$, Using $\text{T}_{\text{E}}\text{X}$ to write polyphonic or instrumental music* [9], con una extensión de más de cien páginas. Mi contribución no irá más allá pues de una pequeña introducción a este documento. Trataré de mantener un cuidadoso equilibrio entre la descripción de las posibilidades de MusiX $\text{T}_{\text{E}}\text{X}$ y los tecnicismos del código, de modo que el lector pueda hacerse una idea más precisa de qué supone la codificación de una partitura.

3. La escritura musical es bidimensional

Esta afirmación es fundamental para asimilar con rapidez la manera de codificar la notación musical utilizada por MusiX $\text{T}_{\text{E}}\text{X}$. En la música puramente monódica una partitura es estrictamente lineal, desarrollándose en el tiempo. Sin embargo en la polifonía o en la música instrumental (con más de un instrumento o un único instrumento de teclado: piano, órgano, etc.) una partitura posee dos modos de «evolucionar»: uno vertical en el que se recorren las distintas voces, pentagramas o instrumentos, y uno horizontal que sigue la secuencia temporal de la música. La alineación vertical entre dos o más figuras que representan sonidos simultáneos ha de ser perfecta, de modo que una partitura con diversos pentagramas debe ser pensada, con vistas a su codificación, como una pila vertical conjunta que evoluciona horizontalmente.

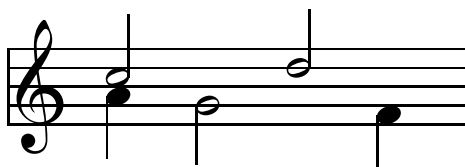
Esta forma de mirar una partitura es además la habitual para los músicos que deben aprehender, visual y mentalmente, en un golpe, porciones verticales, una tras otra. Veamos un ejemplo: el comienzo de la sonata para piano en

do mayor KV. 545 (*sonata facile*) de Wolfgang Amadeus Mozart²:



Observe la perfecta alineación vertical. Comenzamos con un do blanca en la mano derecha, que se encuentra exactamente sobre la primera de las corcheas del primer grupo barrado en la mano izquierda; le siguen dos negras (mi, sol) sobre la primera y tercera corcheas del segundo grupo barrado (una blanca dura lo mismo que cuatro corcheas, una negra lo mismo que dos). En el segundo compás observe el grupo de dos semicorcheas (grupo con dos barras): la primera se alinea verticalmente con una corchea, la presencia de la segunda semicorchea produce un espacio mayor de separación entre los dos grupos barrados de la mano izquierda.

Como se puede observar en el ejemplo las notas de más duración son seguidas de un espacio de mayor longitud: es una tradición útil ya que el tamaño de los espacios ayuda al músico a captar inmediatamente la duración. El espacio posterior a una nota puede no depender de su duración, sino de las notas adyacentes, como ocurre en el siguiente ejemplo³:












donde se observa que el espacio posterior a cada blanca es el correspondiente a una negra, ya que cada figura posterior a una de estas blancas «entra» en la mitad de su duración.

Los elementos básicos están sobre el tapete: alineación vertical de notas simultáneas y espaciado proporcional al tiempo entre dos figuras consecutivas. ¿Cómo consigue MusiX_{TeX} conjugar estos elementos? El usuario debe dividir

²Este ejemplo aparece como primer ejemplo de codificación en [9]; lo he tomado de la edición de Füssl y Scholz, publicada por Real Musical en 1981, y, sin duda, en la elección también ha influido su carácter *facile* o, de otro modo, la circunstancia de ser uno de los pocos fragmentos de las sonatas mozartianas que puedo disfrutar interpretando.

³Tomado también de [9]

| Secuencia ⁴ | Espaciado | Previsto para |
|---|---------------------------|---|
| <code>\notes ... \enotes⁵</code> | <code>2\elemskip</code> |  |
| <code>\notesp... \enotes</code> | <code>2.5\elemskip</code> |  |
| <code>\Notes ... \enotes</code> | <code>3\elemskip</code> |  |
| <code>\Notesp... \enotes</code> | <code>3.5\elemskip</code> |  |
| <code>\NOTes ... \enotes</code> | <code>4\elemskip</code> |  |
| <code>\NOTesp... \enotes</code> | <code>4.5\elemskip</code> |  |
| <code>\NOTes ... \enotes</code> | <code>5\elemskip</code> |  |
| <code>\NOTesp... \enotes</code> | <code>5.5\elemskip</code> |  |
| <code>\NOTEs ... \enotes</code> | <code>6\elemskip</code> |  |

Cuadro 1. Secuencias fundamentales para la codificación de notas o método de selección del espaciado entre notas

la partitura en bloques verticales que serán codificados utilizando cada uno de ellos un tipo de espaciado entre notas. Como comprobaremos inmediatamente la posible división en bloques verticales no es única y la elección realizada puede afectar a la simplicidad del código.

El espaciado entre notas se selecciona mediante la elección de una de las secuencias fundamentales para la codificación de notas, que son las indicadas en el cuadro 1.

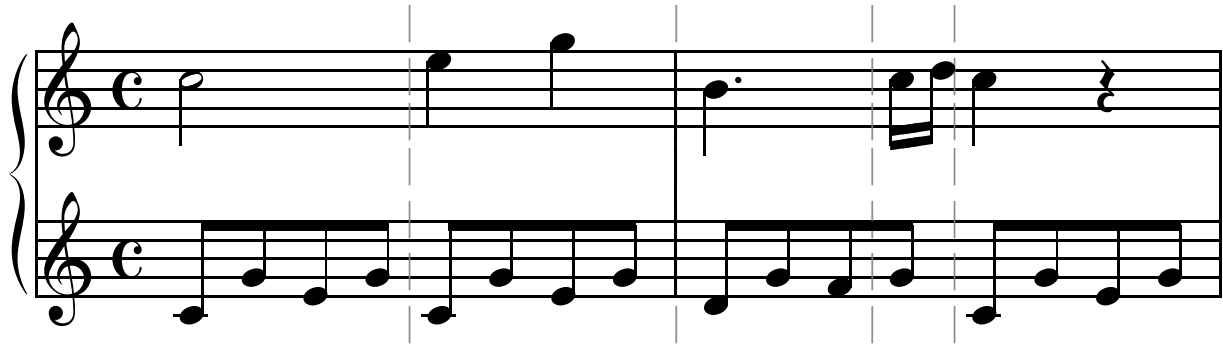
Volviendo al fragmento mozartiano, la figura 1 muestra una posible división en bloques verticales y el código correspondiente (naturalmente, exceptuadas las líneas discontinuas que indican la división).

Observemos que en todo el fragmento el espaciado inferior lo dan las corcheas de la mano izquierda, excepto en el segundo tiempo del segundo compás, donde aparece un grupo de dos semicorcheas, lo que justifica la introducción de un bloque vertical que lo contenga. Todos los comandos interiores a las secuencias `\Notes... \en` son los que ubican las notas y serán comentados posteriormente. En este punto interesa destacar la secuencia:

$$\backslashql\{1*n\}$$

⁵Existe también la secuencia `\znotes... \enotes`, prevista para notas no espaciadas.

⁵El comando `\enotes`, abreviatura de «end notes», puede ser sustituido por `\en`.



```

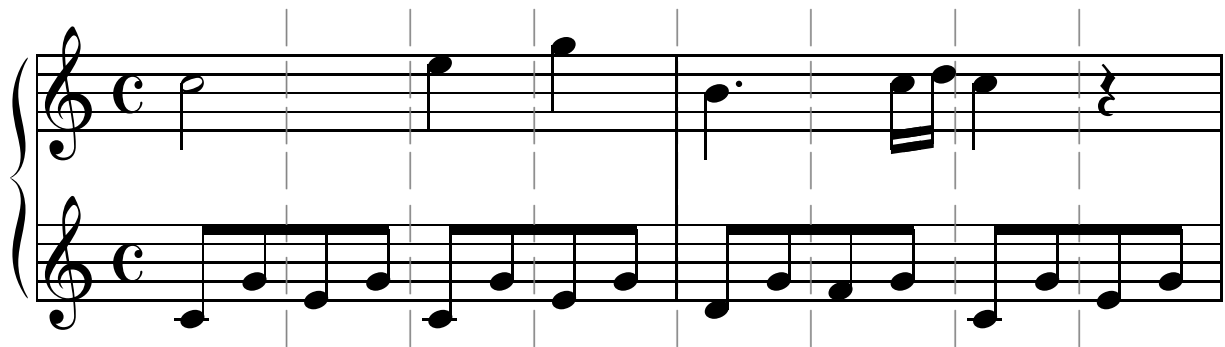
\begin{music}%(Preámbulo, que detallo más tarde)
\startextract\addspace{\afterruleskip}
%Primer bloque o secuencia \Notes...\en:
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\hl j\en
%La barra | cambia de pentagrama. La codificación es:
%pentagrama inferior | pentagrama superior. Sigue el segundo bloque:
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\ql{1*n}\en
\bar % Barra de compás (\bar). Tercer bloque:
\Notes\ibu0f0\qb0{dgf}|\qlp i\en
%Cuarto bloque:
\notes\tbu0\qb0g\sk|\ibb1k3\qb1j\tb1\qb1k\en
%Quinto bloque:
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\ql{j*}\qp\en
\endextract\end{music}

```

Figura 1. Primera división en bloques verticales

El comando `\ql` indica una negra con la plica hacia abajo. Las letras `l` y `n` indican las notas *mi* y *sol* (cuarto espacio y primer espacio adicional en clave de *sol* en segunda). ¿Qué significa el asterisco en `\ql{1*n}`? Se trata de una nota «fantasma», es decir, el espacio correspondiente a una nota⁶. Este espacio es el correspondiente a una corchea, ya que el comando está incluido en una secuencia `\Notes. .\en`. Puesto que la primera negra posee la duración de dos corcheas necesitamos introducir este espacio adicional, de modo que la segunda negra (`n=sol`) coincida con la tercera corchea del grupo barrado de la mano izquierda. Algo semejante ocurre en el segundo compás con las secuencia: `\ql{j*}\qp`, que introduce una negra (`do=j`), un espacio (de corchea) y

⁶Una alternativa a la codificación `\ql{1*n}` es `\ql 1\sk \ql n`, donde `\sk` representa el espacio de una nota (`skip`).



```

\begin{music}%(Preámbulo)
\startextract\addspace{\afterruleskip}
\Notes\ibu0f0\qb0{cg}|\hl j\en
\Notes\qb0e\tbu0\qb0g|\en
\Notes\ibu0f0\qb0{cg}|\ql l\en
\Notes\qb0e\tbu0\qb0g|\ql n\en \bar
\Notes\ibu0f0\qb0{dg}\qb0f|\qlp i\en
\notes\tbu0\qb0g|\ibbl1k3\qb1j\tbl1\qb1k\en
\Notes\ibu0f0\qb0{cg}|\ql j\en
\Notes\qb0e\tbu0\qb0g|\qp\en
\endextract\end{music}

```

Figura 2. Segunda división

un silencio de negra (`\qp`, «quarter pause»).

La figura 2 muestra otra división y el código que le corresponde. En esta segunda forma de división en bloques tenemos un código con más secuencias fundamentales (`\Notes . . . \en`), pero se evita la necesidad de introducir espacios manualmente (mediante el asterisco).

Parece claro, pues, que lo que marca el comienzo y final de un bloque debe ser el menor intervalo de tiempo entre el ataque de dos notas consecutivas: cuando este tiempo aumenta o disminuye es hora de cerrar el bloque actual e iniciar uno nuevo.

4. MusiX_TE_X es un sistema de tres pasos

Es bien conocido que T_EX compone los párrafos con la ayuda del «glue»: un espacio elástico que se introduce entre palabras consecutivas. El complicado algoritmo de construcción de párrafos de T_EX (véase [5]) decidirá dónde cortar

para formar las líneas y el «glue» se ensanchará o comprimirá en la medida necesaria.

La unidad «indivisible» de texto es la palabra (o fragmentos de ellas en caso de que se utilice la división silábica). En una partitura, sin embargo, la unidad indivisible es el compás. La tradición musical impone que un compás no sea dividido en dos partes por un salto de línea (aunque en contadísimas ocasiones pueda ser inevitable, por ejemplo en cadencias muy largas).

Así MusiX_{TEX} podría haber optado por trabajar en forma similar para componer las líneas en una partitura: introducir la adecuada cantidad de «glue» entre compases. Pero la diferencia con el texto ordinario es sustancial: mientras que en una línea de texto el número de palabras es lo suficientemente grande para que la flexibilidad en la dimensión del «glue» no provoque enormes blancos, en una línea de música el número de compases es muy inferior (a menudo dos o tres).

Para resolver este problema MusiX_{TEX} opta por no introducir «glue» sino una longitud (`\elemskip`) recalculable. En una primera compilación el valor de `\elemskip` está fijo (dependiendo del tamaño elegido para las notas). El espaciado en cada secuencia fundamental `\notes . . . \en` es proporcional a este valor (tal y como se indica en el cuadro 1). Una rutina externa (`MUSIXFLX.EXE`) calculará después cómo modificar el valor de esta longitud en cada línea de la partitura, a fin de que cada una se complete perfectamente. El nuevo valor de `\elemskip`, propio de cada línea, se almacena en un archivo que será leído en una segunda compilación.

Supongamos, como un ejemplo, que deseamos compilar un documento, `Doc.tex`, que contiene algún fragmento de código MusiX_{TEX}; he aquí la secuencia a realizar:

1. compilación: se crea el archivo `Doc.mx1`;
2. ejecución de `MUSIXFLX` sobre `Doc.mx1`: se genera `Doc.mx2`;
3. nueva compilación, que utiliza los nuevos valores de `\elemskip` archivados en `Doc.mx2`.

Es interesante observar los resultados sobre un ejemplo concreto. Utilicemos el comienzo del *Rex tremendæ* del Requiem de W. A. Mozart, en la parte de los trombones tenores. Tras la primera compilación, para la que hemos puesto `\textwidth=10cm`, el resultado es el que muestra la siguiente figura (las dimensiones no coinciden, pues la figura se ha escalado); observe las curiosas figuras volantes:

The image shows a musical score for three staves. The top staff is in treble clef with a common time signature (C). It contains a sequence of notes and rests, with a dynamic marking *f* below it. The middle staff is in bass clef and contains a sequence of notes and rests, with a dynamic marking *ff* below it. The bottom staff is in bass clef and contains a sequence of notes and rests, with a dynamic marking *ff* below it. The score is enclosed in a rectangular box.

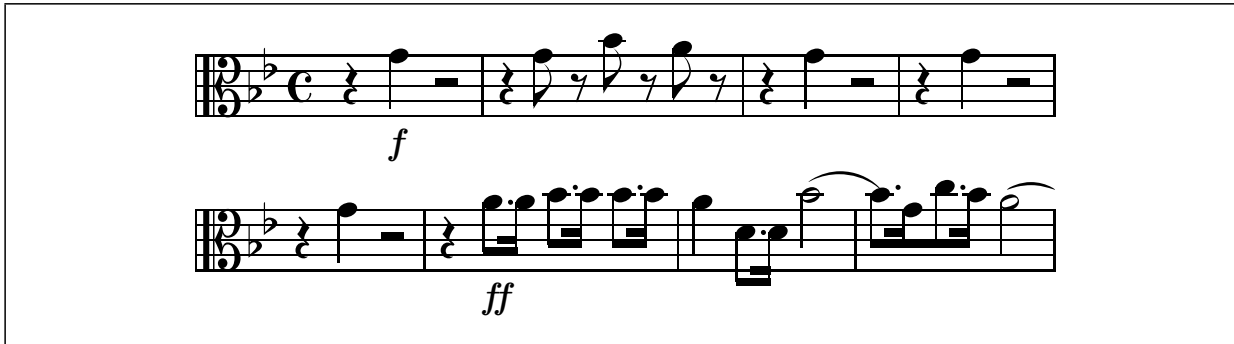
Después de ejecutar MUSIXFLX y de compilar una segunda vez, el resultado es impecable:

The image shows a musical score for three staves, identical to the one above. The top staff is in treble clef with a common time signature (C). It contains a sequence of notes and rests, with a dynamic marking *f* below it. The middle staff is in bass clef and contains a sequence of notes and rests, with a dynamic marking *ff* below it. The bottom staff is in bass clef and contains a sequence of notes and rests, with a dynamic marking *ff* below it. The score is enclosed in a rectangular box.

Naturalmente cualquier cambio del código fuente necesitará, en primer lugar, eliminar el archivo `.mx2`, para, después, reiniciar los tres pasos.

En relación con este problema del espaciado y el sistema de tres pasos otra tradición de la tipografía musical viene a complicar la tarea: la que sugiere que la última página de cualquier partitura musical (extensa) sea completamente llenada. Esto puede implicar la necesidad de aumentar manualmente el número de líneas, para lo que MusiX $\text{T}_{\text{E}}\text{X}$ provee el parámetro `\mulooseness` (análogo al `\looseness` de $\text{T}_{\text{E}}\text{X}$).

Un cambio del valor por defecto (nulo) de este parámetro necesitará reiniciar todo el proceso. Así, incluyendo por ejemplo, `\mulooseness=-1` en el fragmento anterior, borrando `Doc.mx2`, y realizando los tres pasos, obtendríamos:



5. Una rápida mirada a la codificación

La dificultad en la codificación de partituras con MusiX_{TEX} procede fundamentalmente de tres aspectos:

1. la bidimensionalidad ya comentada;
2. la «pasividad» de MusiX_{TEX}: no toma ninguna decisión musical o estética (no contabiliza la duración de las notas, no ubica las barras de compás, no selecciona grupos barrados salvo que se le diga explícitamente, etc.); el usuario debe indicar absolutamente toda la información ¡y es mucha!, lo que, en particular, dificulta enormemente la tarea de corrección del código;
3. los problemas de espaciado entre notas, de rotura de líneas, de páginas, etc.; en particular los espacios espúreos, sobre todo los provocados por finales de línea que no terminan en un comando o en un %, provocan resultados defectuosos en el espaciado.

A continuación se indican los comandos para la escritura de algunos de los elementos musicales más básicos: notas, figuras y acordes.

Notas

En MusiX_{TEX} las notas reciben el nombre abreviado de la notación anglosajona: a para «la», b es la nota «si», hasta g que es «sol». La sucesión alfabética puede ser continuada para nombrar notas de octavas superiores. Se dispone también de letras mayúsculas para notas de octavas inferiores. El cambio de octava puede ser indicado también haciendo preceder el nombre de la nota de uno o más apóstrofos: «'» para elevar una octava, «''» para elevar dos octavas, «'» para descender una octava, etc. Los apóstrofos de transporte por

octavas son acumulativos en cada pentagrama y en cada secuencia fundamental \notes... \en. Para que dejen de actuar se introduce el símbolo ! (véase el fragmento de la página 54).

El fragmento que sigue⁷ muestra por encima (y/o por debajo) de cada nota los diferentes nombres que puede recibir.

The image displays a musical score with three staves. The first staff is a bass clef staff with notes and labels: 'A 'B 'C 'D 'E 'F 'G A B C D E F G H I J and ''a ''b ''c ''d ''e ''f ''g 'a 'b 'c. The second staff is a bass clef staff with notes and labels: K L M N a b c d e and 'd 'e 'f 'g. The third staff is a treble clef staff with notes and labels: 'D 'E 'F 'G and a b c d e f g h i j k. Below the third staff, there are two rows of labels: 'e 'f 'g ''a ''b ''c ''d ''e ''f ''g '''a '''b '''c '''d '''e and l m n o p q r s t u v w x y z.

Paso entre pentagramas e instrumentos

MusiX_{TEX} establece un orden particular en la lectura vertical de una partitura: se codifica desde el pentagrama inferior hacia arriba. En el interior de cada secuencia \notes... \en se pasa de un instrumento al siguiente (es decir al pentagrama inferior del instrumento situado por encima del actual) mediante el carácter «&» o su sinónimo \nextinstrument. Cada instrumento puede tener de uno a cuatro pentagramas, en tal caso el paso de un pentagrama al siguiente se lleva a cabo mediante el carácter «|» o bien mediante su sinónimo \nextstaff. También se puede «saltar» directamente a un pentagrama de un instrumento dado.

⁷tomado, salvo pequeñas modificaciones de formato, de [9].


```
\endextract
\end{music}
```

Un concepto fundamental es el de *figuras no espaciadas*, es decir, figuras que no van seguidas de un espacio, por tanto no modifican la posición en la que se ubicará la figura siguiente. Este tipo de figuras son esenciales en la escritura de música polifónica (en un solo pentagrama) y de acordes. Los comandos para estas figuras no espaciadas se construyen de forma similar a las figuras ordinarias, pero su nombre va precedido de la partícula *z*. He aquí un simple ejemplo de polifonía⁸:



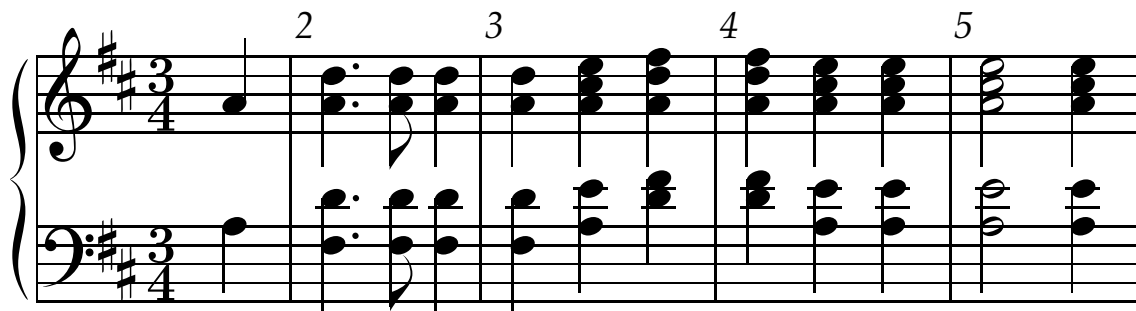
cuya codificación es:

```
\begin{music}
%(Preámbulo)
\startextract %%Primer compás:
\Notes\zhl M\hu b\zhl M\hu a|%%Mano izquierda (inferior)
\zhl d\hu g\zhl d\hu h\en\bar%%Mano derecha
%%Segundo compás
\Notes\zhl N\hu N\zhl M\hu a|
\zhl d\hu i\zhl d\hu h\en
\endextract\end{music}
```

Los acordes se construyen de forma sencilla: se introducen todas las notas que lo conforman, excepto una, mediante comandos (especiales) de figura no espaciada y se termina con la última nota con una figura espaciada. MusiX_{TEX} se encarga de colocar la plica con una longitud adecuada para abarcar todas las notas que componen el acorde. He aquí un ejemplo del uso de acordes⁹:

⁸Coral del Álbum de la Juventud, Robert Schumann, edición Kalmus.

⁹Primeros compases del *Finale* de Papillons, Robert Schumann, Opus 2.



A continuación figura el código que ha generado el ejemplo anterior (tan sólo la parte correspondiente a las notas); observe que he aprovechado para definir un par de comandos que permiten abreviar la escritura: en general, una partitura contiene siempre repeticiones de un fragmento (más o menos exactas, ya sea en las notas o en el ritmo), el uso de comandos propios del autor se adapta magníficamente a esta circunstancia.

%(Preámbulo)

%\ach y \acq: se expandirán en acorde de negras y blancas resp.

```
\newcommand{\acq}[2]{\zq{#1}\ql #2}
```

```
\newcommand{\ach}[2]{\zh{#1}\hl #2}
```

```
\begin{music}\startextract
```

```
\Notes\qa a|\qa{'a}\en\bar%Compás 1
```

```
\Notes\zqp M\qlp d|\zqp{'a}\qlp d\en%Compás 2
```

```
\notes\zq M\cl d\zq M\ql d|%Compás 3, mano izq.
```

```
\zq{'a}\cl d\zq{a}\ql d\en\bar%Compás 3, mano der.
```

```
\Notes\zq M\ql d\zq a\ql e\zq d\ql f|%Compás 4, izq.
```

```
\zq{'a}\ql d\zq{ac}\ql e\zq{ad}\ql f\en\bar%Compás 4, der.
```

```
\Notes\acq df\acq ae\acq ae| %Compás 5
```

```
\acq{'ad}{f}\acq{ac}{e}\acq{ac}{e}\en\bar
```

```
\Notes\ach ae\acq ae|\ach{'ac}{e}\acq{ac}{e}\en
```

```
\endextract\end{music}
```

Naturalmente disponemos de comandos para la escritura de los distintos silencios, así como otro buen conjunto de comandos para notas asociadas a grupos barrados (véase el apartado 6), que no especificaré aquí.

El «preámbulo» musical

Antes de poder codificar notas sobre uno o más pentagramas hemos de indicar a MusiX_{TEX} algunos datos básicos. Es la parte del código que, en los ejemplos anteriores, hemos evitado indicando la marca del %(Preámbulo).

Aquí, obvia quizás insistir, sólo muestro algunas posibilidades en las cuestiones más básicas.

En primer lugar: el número de instrumentos de la partitura se indica mediante `\instrumentnumber{n}`, donde, naturalmente, n será el número de dichos instrumentos, que serán numerados desde 1, el situado en la posición inferior, hasta n . Podemos asignar un nombre al instrumento i -ésimo mediante `\setname{i}{Nombre}`; este nombre aparecerá a la izquierda de cada pentagrama (o grupo correspondiente al instrumento) al comienzo de la pieza.

La armadura se selecciona mediante `\generalsignature{n}`, donde n es un número entero, si es positivo indica el número de sostenidos en la armadura, si es negativo el número de bemoles. El comando anterior selecciona la armadura para todos los pentagramas de todos los instrumentos, aunque también es posible cambiarla para algunos instrumentos. Algo análogo ocurre con la indicación de compás: la selección global se realiza mediante `\generalmeter{M}`, donde M puede ser uno de los comandos predefinidos (`\meterC` para compasillo) o una indicación de fracción del tipo `\meterfrac{3}{4}` (para compás de tres por cuatro).

Finalmente las claves de cada instrumento se seleccionan mediante el comando `\setclef{n}{Claves}`, donde n es el número del instrumento y *Claves* son tantas indicaciones de clave como pentagramas posea el instrumento. Las *Claves* se identifican mediante un número o mediante un comando más descriptivo y fácil de recordar (en inglés, naturalmente¹⁰), por ejemplo: `\treble` para la clave de sol en segunda, `\alto` para do en tercera o `\bass` para fa en cuarta.

Además puede especificarse el tamaño al que se escribirá la partitura (seleccionable para cada instrumento por separado), así como otros muchos elementos.

Posteriormente los comandos `\startpiece` y `\endpiece` marcan el inicio y final de la pieza musical. Una alternativa a éstos viene dada por la pareja `\startextract` y `\endextract`, prevista para encerrar pequeños fragmentos musicales (como los de los ejemplos que he incluido) en lugar de piezas largas.

El entorno `music` es específico para la escritura de partituras en documentos \LaTeX (`MusiXTeX` está pensado para poder trabajar bajo \TeX).

¹⁰En la cuestión del idioma `MusiXTeX` sigue unas pautas un tanto atípicas ya que contiene comandos de nombres en inglés, en francés, la lengua materna de Taupin, e italiano, idioma usual en la música.

6. Grupos barrados

Los grupos barrados constituyen uno de los elementos de la tipografía musical más difíciles de codificar con MusiX_{TEX}. Sin pretender entrar en los detalles técnicos voy a presentar lo imprescindible para obtener una idea de la dificultad, las posibilidades y la belleza del resultado. En realidad MusiX_{TEX} proporciona dos formas de codificar grupos barrados: una manual (la más difícil) y una automática (que tiene restricciones de uso).

En la codificación manual el usuario debe precisar, para cada grupo, los siguientes elementos: un número de referencia, la pendiente de la barra (expresada como un múltiplo de una pendiente base, que es el 5%), la posición en la que se inicia (indicando una nota), el inicio del grupo, las figuras asociadas a él y el final del grupo. ¡Demasiada información! Además los comandos para figuras asociadas a un grupo barrado son distintos de los que ubican figuras ordinarias; en particular deben contener el número de referencia del grupo al que se asocian y ello porque podemos tener varios grupos barrados abiertos simultáneamente.

El siguiente fragmento¹¹ puede dar una buena idea de la dificultad:



El código que muestro a continuación vuelve a recurrir (parcialmente, para ilustrar ambos modos de proceder) a comandos propios del usuario:

```
\newcommand{\BB1}[5][0]%
  {\ibbl{#1}{#2}{#3}\qb{#1}{#4}\tqb{#1}{#5}}
\newcommand{\BBu}[5][0]%
  {\ibbu{#1}{#2}{#3}\qb{#1}{#4}\tqh{#1}{#5}}
\startextract
\notes\ibbl0L1\qb0{JNM}\tb10\qb0N\ibbl0M0\qb0{LNM}\tqb0N|%
\BBu[1]{h}{-1}{jed}{e}\BBu[1]{c}{0}{ced}{e}\en
\notes\BB1 L1{JNM}N\BB1{M}{0}{LNM}{N}|%
```

¹¹Preludio n. 2 BWV 847, de *El clave bien temperado*, J.S. Bach.

```
\BBu[1]{h}{-1}{jed}{e}\BBu[1]{c}{0}{ced}{e}\en
\endextract
```

Comentemos brevemente alguno de los comandos que aparecen en el fragmento de código anterior. `\ibbl 0L1` inicia un grupo de semicorcheas (de ahí la doble bb) hacia abajo (1), con identificador 0, iniciándose la doble barra tres espacios (del pentagrama) por debajo de la nota L (tercer espacio en clave de fa), con pendiente $1 \times 5\%$. A este comando (el que genera el primer grupo barrado del pentagrama inferior) le sigue el código `\qb0{JNM}` que coloca, asociadas al grupo barrado 0, las notas (negras, «q») JNM; a continuación aparece: `\tbl0\qb0N` que termina el grupo 0, hacia abajo (un grupo hacia abajo puede concluir hacia arriba), con la figura (negra) en la nota N. ¿Le resulta complicado? Comandos similares a los utilizados son: `\ibu`, `\ibl` para grupos de corcheas, `\ibbbu`, `\ibbb1` para grupos de fusas, etc.; `\tbu` para terminar un grupo por encima de la nota, y algunas abreviaturas, por ejemplo: `\tqh in` (utilizada en el código anterior) es una abreviatura para `\tbu i\qb in`.

Veamos el mismo fragmento con grupos automáticos:



```
\begin{music}
\setstaf12\generalmeter{\meterC}
\generalsignature{-3}
\setclef1{\bass\treble}\startextract
\notes\Qqbb1 JNMN\Qqbb1 LNMN\Qqbb1 JNMN\Qqbb1 LNMN|
\Qqbbu jede\Qqbbu cede\Qqbbu jede\Qqbbu cede\en
\endextract\end{music}
```

Como puede observar se trata de comandos preparados para grupos barrados de cuatro semicorcheas (`\Qqbbu` y `\Qqbb1`), pero también disponemos de comandos para grupos de dos o tres corcheas o semicorcheas (`\Dqbu`, `\Tqb1`, `\Qqbu`, `\Dqbbu`,...).

En realidad estos comandos son abreviaturas de unos más básicos, que, digamos, generan grupos barrados semi-automáticos: `\Ibu nABp` iniciaría un

grupo hacia arriba, identificado por el número n , que se extiende el espacio ocupado por p figuras y cuya primera y última notas son A y B .

Con más o menos esfuerzo MusiX_{TE}X permite codificar grupos barrados que se inician en un compás o en una línea y terminan en el o la siguiente, grupos de duración variable (por ejemplo que se inician como un grupo de corcheas y terminan siendo de semicorcheas), así como grupos que pasan de un pentagrama a otro.

7. Ligaduras

Las ligaduras representan uno de los elementos de la tipografía musical más complicados y, en consecuencia, donde MusiX_{TE}X muestra su mayor debilidad. Con todo se defiende bastante bien (aunque no con sencillez) y, como ya comenté en la breve historia inicial, otros paquetes han venido a fortalecer esta flaqueza.

En la nomenclatura musical se distinguen tres tipos de ligaduras: las de prolongación, articulación y fraseo. En español no hay nombres diferentes para ellas, aunque en inglés existen dos términos: *tie* para las de prolongación y *slur* para el resto.

MusiX_{TE}X divide las ligaduras en *simples* y *compuestas*. Las simples son caracteres de una fuente, mientras que las compuestas se forman concatenando varios caracteres. Las primeras se seleccionan internamente cuando la ligadura es suficientemente corta y con pendientes no excesivas. La codificación de las ligaduras es similar a la de las barras: es preciso indicar un número que las identifica, su inicio (antes de la figura en la que comienzan) la posición en la que se inician (una nota) y el final (también antes de la figura). Además MusiX_{TE}X dispone de comandos para modificar el aspecto de una ligadura, permitiendo controlar su altura, así como las pendientes inicial y final.

Me limitaré a mostrar un ejemplo¹². En este fragmento que sigue se ha modificado la configuración de la ligadura de fraseo más larga (la que comienza en el tercer compás), debiendo modificar, además, la separación vertical entre los dos pentagramas para evitar la superposición de dos ligaduras. El código (sólo en lo que concierne a las ligaduras) figura después del fragmento.

¹²Tomado de una adaptación para piano a cuatro manos del famoso fragmento *El Cisne*, del Carnaval de los Animales de Saint-Saëns.



%Compás 1. Se inicia (\islur) una ligadura (0), hacia abajo (d) en la nota g (sol)

%Se termina con \tslur 0b. La segunda es similar

\NOTes\islurd0g\qu{gf}\tslur 0b\qu b\islurd0e\qu{ed}%

\tslur0{'g}\qu{g}\en\bar

%Compás 2. Ligadura de prolongación. Observe que sólo se necesita indicar

%la nota al inicio. \itied la inicia, \ttie la termina

\NOTes\islurd 0a\itied 1a\hu a\en

\notes\ttie1\ibu0a1\qb0a\tbu0\qb 0b\tslur 0c\hu c\en...\bar

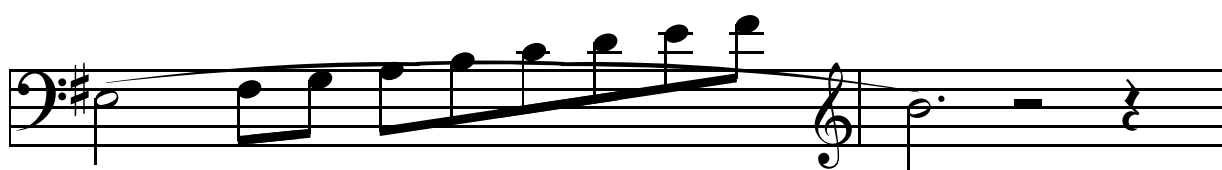
\NOTes\isluru 0L...\en%Compás 3, se inicia una nueva ligadura

\Notes\ibl0N2...\en ...\bar%Se termina, modificando sus propiedades

%\curve 863, indica variación vertical de 8, pendientes inicial (6) y final (3):

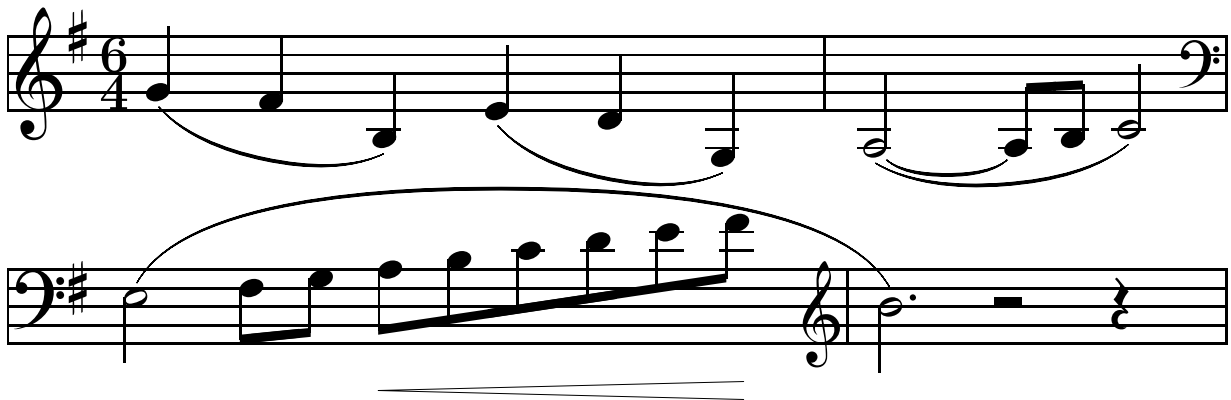
\NOTes\curve 863\tslur0i\hlp i\hpause\qp\en

La modificación de las propiedades de la ligadura es imprescindible, ya que MusiX_{TEX} no considera, en ningún caso, la posible superposición de elementos en una partitura. El resultado sin las oportunas modificaciones es nefasto:



Podemos comparar el resultado anterior con el que proporciona el paquete musixps, incluido a continuación; observe que el resultado es muy superior, sobre todo en la tercera ligadura (entre la y do). Las ligaduras que proporciona musixps son siempre simétricas lo que, independientemente de criterios estéticos, es una limitación¹³).

¹³El uso de musixps incrementa las posibilidades de sobrepasar la capacidad de registros de $\text{T}_{\text{E}}\text{X}$, véase la página 64.



8. ¿Qué más puede hacer MusiX_TE_X?

Simplemente *todo...* o casi todo. Dispone de una multitud de signos musicales, más que suficientes, salvo, quizás, en lo que respecta a la música contemporánea. La figura 3 contiene una buena muestra de ellos. Otros paquetes, que son parte de MusiX_TE_X, digamos extensiones, ofrecen símbolos para música de instrumentos de percusión, gregoriano, indicaciones para cuerdas, tablaturas para guitarra, etc.



Figura 3. Plétora de símbolos de MusiX_TE_X. Algunos de los mostrados dependen de paquetes adicionales que se incluyen en la distribución usual de MusiX_TE_X. Existen otros símbolos adicionales que no muestro

A continuación destaco algunas otras posibilidades de MusiX_{TEX}.

- Varios tamaños de música: se dispone de cinco tamaños, que pueden ser seleccionados independientemente para cada instrumento. Además de esto, por supuesto, dispone de la posibilidad de introducir notas de pequeño tamaño (apoyaturas, ornamentos,...).
- En piezas orquestales o de cámara permite seleccionar los instrumentos por separado, aunque esta selección no es totalmente automática.
- Contiene herramientas suficientes para el control de la composición tipográfica y la paginación.
- Transporte automático, en un 99 %: no selecciona automáticamente la nueva armadura, pero esto es un mal menor.

Transporte

El transporte requiere un comentario más detallado. Aparte de la necesidad de modificar manualmente la nueva armadura, un mal realmente menor, la cuestión del transporte de las alteraciones accidentales es importante. El transporte de estas alteraciones depende de la intención del compositor, por lo que MusiX_{TEX} no realiza por defecto dicho transporte. Existe sin embargo la opción de codificar con «alteraciones relativas» o «absolutas», lo que, en el primer caso, dificulta un tanto la codificación pero permite el transporte de estas alteraciones. Para no extenderme más, mostraré un par de ejemplos.

A continuación tenemos la escala melódica de la menor:



codificada como:

```
\relativeaccid\smallmusicsize
\nobarnumbers\setclef1\treble
\startextract% ^ significa sostenido; = significa becuadro
\notesp\qa{abcde^f^g'a'=g=fedcba}\en
\endextract
```

Ahora transportamos ambas escalas a re menor y do sostenido menor:



El código que ha generado ambos transportes es el siguiente:

```
\relativeaccid\nobarnumbers
\setclef1\treble\smallmusicsize\generalsignature{-1}
\startextract\transpose=3
\notesp\qa{abcde~f~g'a'=g=fedcba}\en
\endextract
\generalsignature{4}
\startextract\transpose=2
\notesp\qa{abcde~f~g'a'=g=fedcba}\en
\endextract
```

Observe que el transporte se realiza mediante la asignación de un valor al registro `\transpose` que especifica el número de posiciones en un pentagrama que se desplazarán las figuras (dicho sea de paso: `{'c}` es equivalente a `{\transpose=7 c}`, mientras que `{'c}` lo es a `{\transpose=-7 c}`). Observe también que se han tenido que añadir las nuevas armaduras (mediante `\generalsignature{-1}` y `\generalsignature{4}`).

La presencia del comando `\relativeaccid` es esencial en el proceso anterior. De no haberlo incluido (o de haberlo sustituido por su opuesto `\absoluteaccid`) el resultado del segundo código habría sido el siguiente:



Como se ve las alteraciones no se modifican.

Principales limitaciones (si lo son)

- Problemas de capacidad (número de registros de $\text{T}_{\text{E}}\text{X}$, sobre todo de `\dimen` y `\skip`). Estos problemas pueden aparecer con una cierta fre-

cuencia cuando se compila con \LaTeX , debido a la multitud de registros que éste introduce por sí sólo. Mi experiencia es que, incluso con \LaTeX , los problemas aparecen únicamente cuando el usuario define una gran cantidad de nuevas longitudes.

- Se pueden codificar partituras para un máximo de seis instrumentos, así como de seis grupos barrados y ligaduras simultáneos. Existen extensiones (paquetes) que permiten ir hasta 9 y 12, para todos esos elementos, aunque aumentan los problemas de capacidad.
- Se dispone de un número máximo de cuatro pentagramas por instrumento (realmente esto no parece una limitación).
- El tratamiento de los textos (letras) en la música vocal es imperfecto: no hay muchos automatismos y requiere un inusual ir y venir entre el código fuente y la salida. Esta circunstancia mejora enormemente con el paquete `musixlyr`.

9. Perspectivas

Para terminar incluyo algunos comentarios muy breves acerca de otras herramientas, citadas en la historia inicial, así como algunos sitios de Internet donde encontrar mucha más información.

Vías de simplificación

`MusiX \TeX` es de uso complicado principalmente porque la cantidad de información que se le debe suministrar es enorme; o, de otra forma, porque no toma ninguna decisión por sí sólo. Sin duda a esta dificultad contribuye la visión bidimensional de la escritura musical.

Estas dos obstrucciones son aliviadas por algunos paquetes y/o precompiladores. ¿Es posible la codificación unidimensional (horizontal), es decir, por pentagramas o incluso por voces? La respuesta es afirmativa a través de dos opciones:

- el paquete `musixser`, que trabaja junto a `MusiX \TeX` y, por tanto, no reduce la cantidad de información a suministrar, aunque simplifica la codificación al permitir que se realice por voces;
- el precompilador `PMX`.

M-TX es un precompilador para PMX (podríamos llamarlo por tanto un «pre-pre-compilador»), que tiene en cuenta la dificultad de la música vocal y utiliza musixlyr. M-TX simplifica considerablemente el código, aunque naturalmente hereda las limitaciones, algunas severas, de PMX, entre las que puedo citar:

- máximo de 12 pentagramas simultáneos;
- máximo de dos voces por pentagrama;
- máximo de 600 compases;
- salida con un máximo de 20 páginas.

A pesar de estas limitaciones PMX y/o M-TX pueden ser enormemente útiles en muchas situaciones. Como ilustración incluyo a continuación el código (completo) para M-TX correspondiente al fragmento de la sonata de Mozart, de modo que puede ser comparado con los mostrados en las figuras 1 y 2.

Style: Propio

Propio: Voices RH LH; Clefs G G; Continuo

%% w120m

```
c2+      e4      g      | b4d-  c1d  c4      r4  |
c8 g+ e g c- g+ e g | d g f g      c- g+ e g |
```

Observe que la duración de una figura se prolonga a las siguientes hasta que se introduce una nueva duración (así c2 e4 g representan un do blanca, seguido de dos negras: mi y sol). Así mismo, observe que no se especifica ningún espaciado entre notas, ni se necesitan introducir «notas fantasma», M-TX se ocupa de contabilizar la duración y colocar las figuras en el lugar adecuado. Tampoco es necesario indicar los grupos barrados de la mano izquierda: M-TX posee unos procedimientos internos para agrupar notas. El código %% w120m es la forma de indicar que la anchura del texto para el fragmento musical será de 120 mm. No muestro la salida obtenida puesto que, finalmente, es (casi) idéntica a la de la página 46, ya que es producida por MusiX_{TEX}.

El precompilador PMX permite obtener además un archivo en formato MIDI, un tanto pobre eso sí, ya que no incluirá apoyaturas, ornamentos ni el efecto de los reguladores (crescendos o decrescendos).

LilyPond

De haber comenzado a interesarme en estos días por la escritura de partituras musicales por medio de alguna herramienta más o menos cercana a $\text{T}_{\text{E}}\text{X}$, teniendo en cuenta lo que hoy conozco, probablemente hubiera optado

directamente por aprender LILYPOND, no MusiX_{TEX}. LILYPOND es un lenguaje, junto con su intérprete, que a partir de un fichero fuente en dicho lenguaje (sólo texto), genera un fichero en código _{TEX} (con abundante código PostScript en su interior). El código _{TEX} generado no es código MusiX_{TEX}: LILYPOND es totalmente independiente de MusiX_{TEX}. LILYPOND fue diseñado inicialmente para Linux, pero existe ya para Windows. Viene acompañado de un script que permite realizar todas las tareas en una sola ejecución: a partir de un código fuente, el script genera, por defecto, archivos log, dvi, ps y pdf, así como, si se le indica, un fichero MIDI, a partir del cual podemos escuchar la música codificada en el archivo fuente (con limitaciones similares a las comentadas para PMX).

LILYPOND utiliza sus propias fuentes PostScript de símbolos musicales y proporciona un completísimo abanico de posibilidades en la escritura de partituras. Es un trabajo (gratuito) digno de elogio (aunque lo mismo es cierto para todo el software comentado en este artículo).

El uso de LILYPOND es bastante sencillo en tanto no sea necesario cambiar el comportamiento por defecto (LILYPOND, como PMX, toma muchas decisiones por sí mismo); en caso contrario la dificultad crece rápidamente, fundamentalmente porque es necesario recurrir a variables y sentencias derivadas del lenguaje Scheme (el lenguaje en que LILYPOND está escrito).

Como ilustración de LILYPOND voy a mostrar ¡otra vez! el fragmento de la sonata mozartiana, así como su código:

Piano

Observe que las fuentes de LILYPOND no son las de MusiX_{TEX}, en particular las líneas del pentagrama son mucho más gruesas que las producidas por MusiX_{TEX} (aunque ¡hasta esto es configurable!). Ahora observe el código, que posee un cierto parecido con el de M-TX:

```
\score{
  \notes{
    \time 4/4
```

```

\context PianoStaff
  \property PianoStaff.instrument = "Piano "
  <<\new Staff {\relative c''
      {c2 e4 g b,4. c16 d c4 r4}}
    \new Staff {\clef treble \relative e'
      {c8 g' e g c, g' e g d g f g c, g' e g}}
  >> }
\paper{linewidth = 110.\mm }
\midi{ }
}

```

He aquí algunos comentarios sobre el código anterior. La duración de las notas sólo es necesaria cuando cambia; los delimitadores <<...>> indican que lo que encierran se produce simultáneamente (es decir, se alinea verticalmente). El apóstrofo ' indica una octava por encima, mientras que la coma (,) baja una octava. La notas encerradas entre llaves después de un `\relative n` se ubican relativamente a la nota *n* y de forma que el intervalo entre cada nota y la siguiente es como máximo una cuarta. La sentencia `\page{...}` indica que se realice salida en papel (ps, pdf), mientras que `\midi{...}` indica una salida en formato midi, para la que se puede especificar indicación de la velocidad (por ejemplo, `\tempo 4=90`).

Por si el fragmento anterior resulta demasiado fácil, me permito incluir el resultado que produce LILYPOND para el fragmento de Saint-Saëns de la página 60, donde se puede observar la calidad de las ligaduras:

El código correspondiente es:

```

\score{
  \notes{
    \time 12/8

```

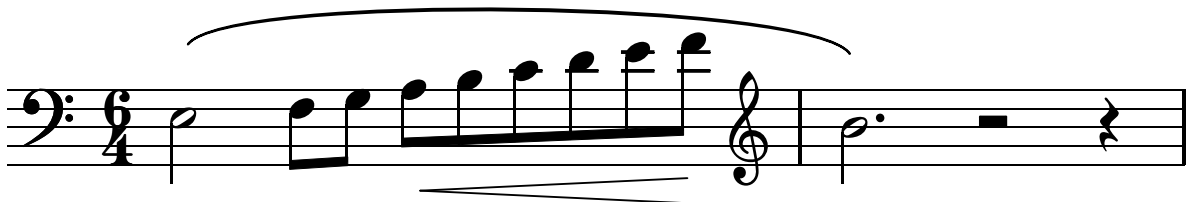
```

\relative g' {g(f b,) e( d g,) a2(~a8 b c2.)}
\clef bass
\property Voice.PhrasingSlur \override #'beautiful = #10.0
e2\( f8 g a[ b c' d' e' f']
\clef treble
b'2.\) r2 r4
}
\paper{indent = 0.\mm
        linewidth = 110.\mm }
\midi{}
}

```

LILYPOND crea internamente los grupos barrados (como en el ejemplo anterior de Mozart) y posee numerosas herramientas para alterar los modos de hacerlo: una forma manual de especificar estos grupos es encerrar las notas entre corchetes [...], como hemos hecho en el código anterior.

Observe la simplicidad en la introducción de las ligaduras de prolongación (mediante el carácter ~) y las de articulación (mediante los paréntesis). Las de fraseo son igualmente simples, mediante \(. . .\), pero suelen requerir un mayor control manual, en este caso alterando el valor de la propiedad `beautiful` del objeto `PhrasingSlur` (que pertenece al «contexto» `Voice`). Es importante señalar que, en caso de no haber modificado manualmente dicha propiedad, LILYPOND habría evitado la colisión de la ligadura con las figuras, exactamente el resultado sería:



que es quizás estéticamente inferior al anterior, pero, en todo caso, mejor que el de la página 61.

Para saber más

En las siguientes direcciones puede encontrar mucha más información, ejemplos, etc.

La página principal en torno a MusiX_{TEX} y sus acompañantes es la página del denominado *Werner Icking Music Archive*:

<http://icking-music-archive.org/>

En ella encontrará casi todo el software, que puede ser descargado, así como una buena colección de partituras, comentarios y enlaces. También podrá subscribirse a la lista de correo electrónico en torno a los sistemas de edición electrónica de partituras musicales; la lista es bastante activa (con una media de ¡al menos dos mensajes diarios!) y se ocupa de una gran variedad de problemas, no sólo en torno a «la compañía MusiX \TeX », sino incluso de cuestiones legales en la edición de partituras, notación y terminología musicales, etc.

A continuación listo algunos otros sitios de interés en Internet; muchos de ellos son los oficiales de las herramientas indicadas.

- De LILYPOND:
<http://lilypond.org/web/>
- De NOTEEDIT:
<http://rnvs.informatik.tu-chemnitz.de/~jan/noteedit/noteedit.html>
- De ROSEGARDEN:
<http://www.all-day-breakfast.com/rosegarden/>
ROSEGARDEN es un programa bajo licencia GNU para Linux enormemente atractivo (pero que yo desconozco). Se trata de un secuenciador de audio y midi, capaz de editar partituras y, lo más interesante aquí, capaz de exportar a PMX y a LILYPOND.
- Para aprender acerca del nuevo lenguaje MusicXML consulte:
<http://www.musicxml.org/xml.html>
- Todo sobre fuentes (tipos), incluyendo musicales, en
<http://cgm.cs.mcgill.ca/~luc/fontresearch.html>
- De \TeX TRACE:
<http://www.inf.bme.hu/~pts/texttrace/>

Referencias

- [1] Cascales, Bernardo; Lucas, Pascual; Pallarés, Antonio y Sánchez-Pedreño, Salvador. *L \TeX . Una imprenta en sus manos* ADI, Madrid, 2000.
- [2] Dunker, Rainer. *musixser* — a MusiX \TeX extension package for voice-wise music coding. 2003

- [3] Dunker, Rainer. *musixlyr* – a MusiX \TeX extension package for lyrics handling. 2003
- [4] Jalbert, François. *M μ \TeX . User's guide*. 1989
- [5] Knuth, Donald. *Digital Typography*. Vol. 78 de CSLI Lecture Notes, Stanford University, 1999.
- [6] Mitchell, Ross; Egler, Andreas y Taupin, Daniel. *Opus \TeX* . Using \TeX to write professional music. β -Test version 0.84, 1998.
- [7] Nienhuys, Han-Wen; Nieuwenhuizen, Jan; Reuter, Jürgen y Zedeler, Rune. GNU LilyPond. The music typesetter. Documentación distribuida con el programa. 1999–2003
- [8] Taupin, Daniel. Comunicación a la lista de correo de MusiX \TeX . Diciembre 1999.
- [9] Taupin, Daniel; Mitchell, Ross y Egler, Andreas. MusiX \TeX . Using \TeX to write polyphonic or instrumental music. Version T. 111. April, 2003.

La esquina del Socio

La Junta Directiva de CervanTEX

.....

1. Grupos de Trabajo

Coordinadores y grupos de trabajo

El trabajo en CervanTEX se organiza en grupos de trabajo, cada uno de los cuales funciona de forma autónoma, establece sus propios métodos de trabajo y elige un coordinador de entre sus miembros. Todos los socios pueden adscribirse libremente a estos grupos de trabajo, para lo cual basta con contactar con los respectivos coordinadores.

La adscripción a uno o varios de ellos no implica que se tenga que desarrollar una actividad concreta: compartir experiencias, aportar sugerencias o simplemente echar una mano de cuando en cuando son actividades útiles que no requieren un gran esfuerzo y siempre resultan útiles. Toda colaboración siempre es bienvenida.

Linux Coordinador: Enrique Meléndez

Windows Coordinador: Christian Reyes

Miembros: Luis Seidel

MacOS Coordinador: Juan Luis Varona

MetaPost Coordinador: Santiago Muelas

La principal actividad de este grupo es el mantenimiento y desarrollo del entorno gráfico Metagraf, que permite crear archivos metapost con una interfaz de usuario.

Estilos para español Coordinador: Javier Bezos

- Estilo spanish para babel.
- Proyectos en curso: nuevos patrones para la división de palabras en castellano.
- Proyectos futuros: estilos bibliográficos para el castellano.

Preguntas frecuentes (FAQ) Coordinador: Agustín Martín Domingo

Mantenimiento y actualización continua de la lista de preguntas frecuentes.

<http://corbu.aq.upm.es/~agmartin/latex/FAQ-CervanTeX/>

Página WEB Coordinador: Javier Bezos.

Colaboradores: José María Goicolea, Santiago Muelas.

Está alojada en el servidor del departamento de Mecanica de la UPM, en la EUI Caminos, Canales y Puertos. <http://www.cervantex.org>.

Lista de correo Mantenedora por: JoséRa Portillo.

Alojada en los servidores de CICA (Sevilla), accesible en <http://www2.cica.es/Listas/sptex>

2. Noticias

Practical TeX 2004. Conferencia sobre formación t técnicas de usuario. Bajo los auspicios del TeX Users Group (TUG), se celebrará en San Francisco, California, del 19 al 22 de junio de 2004. Practical TeX 2004: Training and Techniques <http://www.tug.org/practicaltex2004>.

Reunión del los grupos de usuarios de TeX. La conferencia TUG 2004 se celebrará este año en Xanthi, Grecia, del 30 de agosto al 3 de septiembre. El título completo es:

International Conference on TeX,
XML and Digital Typography - TUG 2004
TeX in the Olympics!

y es la vigésimo quinta reunión anual y conferencia de los grupos de usuarios de \TeX .

Se celebrará en la Universidad Democritus de Tracia. El tema elegido es \TeX en la era de XML y Unicode, y se centrará en el uso de \TeX en :

- la preparación de documentos multilingües complejos para distintos dispositivos en combinación con XML y Unicode
- Combinación de SVG, XSL-FO y XSLT con las \TeX nologías existentes,
- la preparación de libros electrónicos, y
- la edición de lenguajes raros o desaparecidos (lineal A/B, Maya, Jeroglífico, etc.).

<http://obelix.ee.duth.gr/tug2004/>.

GuIT 2004. La reunión del Grupo de Usuarios Italianos de \TeX (GuIT) se celebrará en Pisa, Italia el 9 de octubre de 2004. Es la primera conferencia desde la formación oficial del grupo. <http://www.guit.sssup.it/GuITmeeting/2004/2004.en.html>.

I Congreso de Tipografía. La ADCV junto con la empresa Cromotex y otros profesionales especializados en tipografía están organizando el primer *CONGRESO DE TIPOGRAFÍA EN ESPAÑA* con los objetivos de crear un espacio que permita intercambiar experiencias y formular estrategias para el desarrollo de la disciplina del diseño, en este caso, a través de la tipografía. Presentar un panorama general del diseño tipográfico que se realiza en el Estado español y dar unas pinceladas sobre la producción internacional. Dar a conocer las investigaciones que sobre tipografía, a nivel teórico y práctico, se están llevando a cabo. Provocar una reflexión sobre el papel cultural de una de las herramientas principales del diseño: la letra. Reflexionar sobre la problemática de la producción tipográfica en España.

3. Información de interés

Cuotas

Las cuotas anuales de los socios numerarios para 2004 se agrupan en las siguientes categorías:

| Tipo de socio | TeXemplares | Juegos de CDs | Cuota (€) |
|-------------------------|-------------|------------------|-----------|
| individual con material | 1 | 1 | 30 |
| individual sin material | 1 | - | 20 |
| estudiante con material | 1 | 1 | 21 |
| estudiante sin material | 1 | - | 12 |
| Colectiva (a) | 1 | (+ 12 euros c/u) | 45 |
| Corporativa (b) | 1 | (+ 12 euros c/u) | 120 |

Son socios numerarios las personas que participan en la asociación, bien de forma activa en los grupos de trabajo o en la Junta Directiva, bien ofreciendo su apoyo sin participar regularmente en los trabajos.

(a) La cuota colectiva está pensada para departamentos y organismos públicos en general. Por cada lote de material, se añaden 12 euros. La cuota básica de 45 euros sólo incluye un TeXemplares por número. Los envíos se hacen conjuntamente a una única dirección postal. No se admiten personas jurídicas como tales y los miembros del colectivo conservan su individualidad.

(b) La cuota corporativa está pensada para editoriales y empresas que trabajan con TeX. Por cada lote de material, se añaden 12 euros. La cuota básica de 120 euros sólo incluye un TeXemplares por número. Los envíos se hacen conjuntamente a una única dirección postal. Sólo personas jurídicas registradas como tales. Derechos adicionales: Una página en TeXemplares al año, folletos (hasta 50 g por sobre) junto con el material enviado a los socios, un enlace en estas páginas.

La fórmula más normal en la mayoría de los casos es la de *individual con material*. La variante sin material está pensada sobre todo para aquellos que ya son miembros de otro grupo de usuarios de TeX y no desean recibir el material por duplicado.

Los socios estudiantes debe acreditar esa situación, y no deberán ejercer ninguna actividad profesional.

Las cuotas de los socios deben ser pagadas por adelantado y valen para un *año natural* (del uno de enero al treinta y uno de diciembre). Los nuevos socios pagarán por el año completo, por lo que puede ser recomendable no asociarse a finales del año o hacerlo ya para el siguiente periodo. El envío de material a los socios que no hayan pagado por adelantado estará vinculado a su disponibilidad.

El pago de las cuotas se puede hacer por transferencia o ingreso en la cuenta de CervanTeX, en CajaMadrid, con los siguientes datos:

Banco: 2038
Oficina: 1526
DC: 91
Cuenta: 6000041089

Son socios colaboradores los que de forma voluntaria realizan una aportación económica anual por un valor de al menos el doble de la cuota anual que le corresponda.

Las comunicaciones a la Junta Directiva pueden dirigirse al secretario, en la actualidad Roberto Herrero, a la dirección secretario@cervantex.org, o a la dirección postal siguiente:

Apartado de correos 60118 AP
MADRID
ESPAÑA

T_EXemplares

Año 5.º, número 6 (extraordinario), segundo trimestre 2004

Índice

- 3 Editorial
- 8 *Bibliografía flexible: el sistema `flexbib`* José M. Mira
- 27 *ƎT_EX, Linux y la Botánica: una excelente combinación* Francisco J. Alcaraz Ariza
- 41 *Edición de partituras* Salvador Sánchez-Pedreño Guillén
- 72 *La esquina del Socio* La Junta Directiva de CervanT_EX