

LaTeX bajo Linux

Tipografía Profesional para Todos

Actualmente, todo el mundo tiene ordenadores personales e incluso el usuario más principiante sabe lo que es un procesador de textos. Pero todo aquél que escribiera un libro en el periodo, digamos, de 1455 a 1985, elaboraba un manuscrito que luego mandaba “a imprenta”. La imprenta era un lugar misterioso de donde, por arte de magia, aparecía un ejemplar impreso donde antes sólo había un montón de cuartillas emborronadas. **POR MIGUEL ATENCIA**

Es difícil entender hoy en día el impacto que suponía para un autor ver su obra impresa. LaTeX es el moderno sustituto del maestro tipógrafo. Con un ordenador personal y una impresora, podemos obtener resultados con el mismo aspecto profesional que proporciona una imprenta, tanto si escribimos un libro de 700 páginas como si hacemos un trabajo escolar que incluye gráficos y fórmulas matemáticas. En lo que sigue, daremos algunas razones por las que resulta preferible utilizar LaTeX en lugar de un programa de autoedición o un procesador de textos. Nuestro objetivo es que todos seamos capaces de obtener resultados brillantes con LaTeX lo antes posible (veréis que no es tan difícil) y posiblemente algún purista de LaTeX se moleste en señalar todos las irregulari-

dades que vamos cometiendo. Si no queréis complicaros la vida, ignorad a los puristas.

El primer documento LaTeX

Lo primero es tener LaTeX instalado para hacer todas las pruebas que vamos sugiriendo y las que se os ocurran. LaTeX y Linux se llevan bien, de hecho, LaTeX nació en el mundo Unix, la especificación es pública y la mayoría de las distribuciones son de código abierto. Por lo tanto, es muy posible que LaTeX ya esté en el sistema. Para estar seguros hagamos una prueba. En la línea de comandos escribimos:

```
$ latex
This is TeX, Version ...
**
```

Si al ejecutar el programa *latex* sale algo parecido a lo de arriba, ya podemos pasar directamente a probarlo. Para salir del prompt con los dos asteriscos, simplemente pulsaremos Ctrl-C. Si no tenemos LaTeX instalado, podemos instalarlo desde el CD de nuestra distro favorita mediante un programa de gestión de software, ya sea rpm, Kpackage o cualquier otro. Como último recurso, se puede descargar de la Comprehensive Tex Archive Network [1] o del mirror español en [2]. Aquí encontraremos la distribución teTeX, que es la más portable de las distintas que existen. También podemos encontrar en CTAN [3] alguna documentación sobre LaTeX, la mayoría en formato... LaTeX. Esto es uno de los principales problemas de LaTeX: en general, la ayuda es bastante

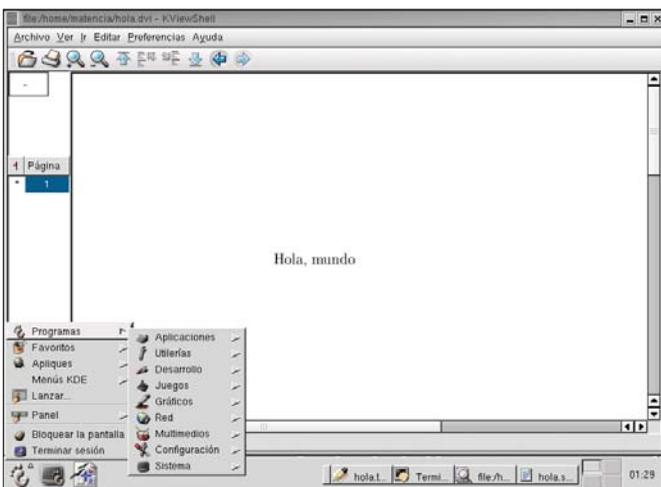


Figura 1: Aspecto de un documento básico en LaTeX. Los márgenes y tipos de letra están ya ajustados a una presentación profesional

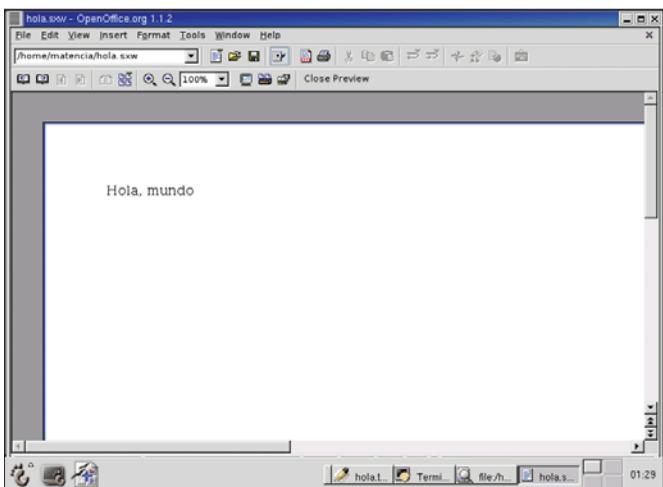


Figura 2: Aspecto de un documento básico en OpenOffice. El resultado es menos profesional

pobre. Con la distribución teTeX se proporciona una ayuda muy básica, en formato info, que puede verse ejecutando *info latex*. Existen excelentes libros sobre Latex en cualquier librería especializada pero no es imprescindible comprarse un libro para obtener resultados brillantes en Latex.

LaTeX no tiene interfaz de usuario, ya que no es un programa interactivo, sino que se ejecuta desde la línea de comandos. Esto será muy fácil de entender para el que haya programado en algún lenguaje (no en un entorno visual): LaTeX es como un compilador. Por tanto, prepararemos el documento LaTeX en nuestro editor preferido. Algunos editores tienen facilidades que ayudan a la escritura de documentos LaTeX. Por ejemplo, si se usa el entorno gráfico KDE, Kate reconoce los comandos LaTeX (aunque no todos) y los representa con distintos colores. De esta forma, podemos detectar errores de sintaxis antes de “compilar”. También se recomienda vim, con la ventaja de que funciona incluso sin entorno gráfico. Muchos intrépidos utilizan emacs o Xemacs, por las posibilidades de personalización que tienen. Si alguien tiene la desgracia de verse obligado a utilizar Windows, la mejor combinación es el editor WinEdit (de pago, aunque muy barato) con la distribución MiKTeX. Si estamos acostumbrados a un procesador de textos, tal como el de OpenOffice, podemos seguir usándolo tranquilamente y disfrutar de las facilidades habituales (el corrector ortográfico, el sistema de autocompletado, etc.). Eso sí, hay que guardar el documento como texto plano y se perderá cualquier formato que se introduzca.

Ya estamos preparados para el primer documento Latex, que será el embrión de nuestra gran obra, el best-seller con que nos haremos famosos. Escribamos en el editor el siguiente texto:

```
\documentclass[a4paper]{book}
\begin{document}
Hola, mundo
\end{document}
```

Al que sea totalmente inexperto en Latex, le recomiendo copiarlo con cuidado, porque si se comete algún error, probablemente Latex lanzará un men-

saje bastante incomprensible. El texto anterior podemos guardarlo, por ejemplo, con el nombre *hola.tex* y ejecutamos desde la línea de comandos:

```
$ latex hola
```

Supongamos que todo ha ido bien e ignoramos la ristra de mensajes que salen. Con esto hemos completado con éxito nuestro primer documento Latex. Ahora bien, ¿qué hemos sacado en claro? Si hubiéramos compilado un programa, habríamos obtenido un ejecutable. Al ejecutar Latex correctamente, obtenemos un fichero DVI, que significa algo así como “fichero independiente del dispositivo”. Un fichero DVI es como un PDF y, al igual que un PDF, necesita un programa para visualizarlo. Todas las distribuciones de Linux decentes incluyen el *xdvi* pero, además, en el menú de gráficos del entorno KDE (o ejecutando *kdvi*) disponemos del KDVI, que resulta mucho más cómodo. En cualquier caso, visualizamos nuestro fichero *hola.dvi* y

observamos el resultado. A la vista de este ejemplo, alguien podría preguntar si no habríamos hecho lo mismo con un procesador de textos, como OpenOffice. La respuesta es sí, pero con algunos inconvenientes. Para empezar, podemos comparar el aspecto de nuestro fichero DVI (Figura 1) con la previsualización de OpenOffice (Figura 2). Esto es subjetivo, pero a mí me parece que el aspecto del documento DVI es mucho más profesional, más parecido a un libro “de verdad”. Naturalmente que en OpenOffice se puede imitar el aspecto de Latex eligiendo los tipos de letra adecuados, márgenes, etc. Pero me parece interesante comparar los documentos que se obtienen con un mínimo de esfuerzo. También resulta ventajoso Latex al usar documentos en texto plano:

```
$ ls -al hola.*
... 5238 sep 19 12:42 hola.sxw
... 74 sep 19 12:33 hola.tex
```

¡El documento Latex es 70 veces más

Listado 1: libro.tex

```
01 % Clase de documento
02 \documentclass[a4paper]{report}
03 % Paquetes adicionales
04 \usepackage[spanish]{babel}
05 \usepackage[latin1]{inputenc}
06 % Fin del preámbulo. Comienza el documento
07 \begin{document}
08 % Página de títulos
09 \title{El maestro de los collares}
10 \author{J.R. Talco}
11 \maketitle
12 % Fin de los títulos. Comienza el texto normal
13 \chapter{De los jovis}
14
15 Este libro trata principalmente de los jovis, y el lector
16 descubrirá en
17 sus páginas mucho del carácter y algo de la historia de este pueblo...
18
19 No obstante, muchos querrán saber desde un principio algo más de
20 este pueblo
21 notable y quizás a algunos no les haya dado la gana comprarse el libro
22 anterior...
23
24 Los jovis son un pueblo complicado pero muy joven, cada
25 vez más
26 numeroso. Amaban la guerra, el follón y la caza...
27
28 % Fin del documento
29 \end{document}
```

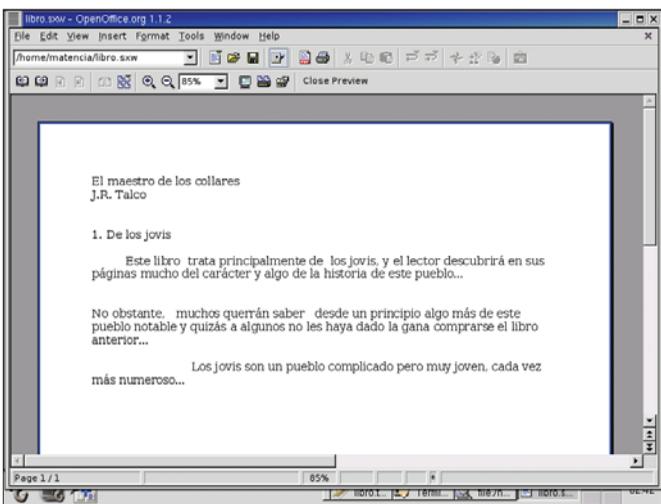


Figura 3: Aspecto del documento en OpenOffice. El resultado impreso arrastrará todas las imperfecciones cometidas al teclear, al ser un programa WYSIWYG

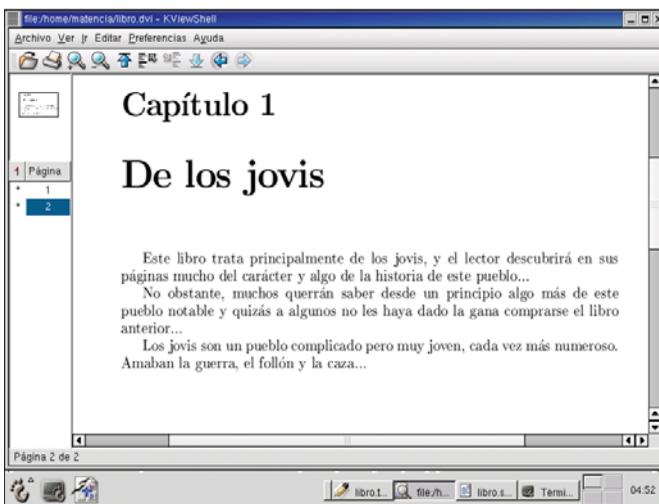


Figura 4: Extracto de un documento completo en Latex, que introduce automáticamente cabeceras de capítulos, página de títulos y demás detalles de formato

pequeño! Baste decir que mi tesis doctoral de casi 200 páginas, con cientos de fórmulas y más de 20 figuras, cabe en un disquete y la escribí en un Pentium 1. Intentad hacer eso mismo con OpenOffice, no hablemos ya de MS Word. Por último, la facilidad de manejo: Latex tiene fama de ser muy difícil de usar, asequible sólo para matemáticos y demás gente de mal vivir. Bueno, ya hemos visto que no es para tanto. Esto es como todo, lo básico e imprescindible es simple, después podemos ir complicándolo todo lo necesario. Incluso podríamos usar el documento que hemos hecho como plantilla y escribir en el “hueco” entre `\begin{document}` y `\end{document}`, sin aprender nada más de Latex. En realidad, esto de usar plantillas hechas por otros es un pecado que, más o menos, cometemos todos los usuarios de Latex. Cuando empecé a usar Latex, me aconsejaron que jamás empezara un documento desde cero, siempre debía partir de una plantilla hecha por otro. He seguido este consejo y no me ha ido del todo mal.

Capítulo 1...

Ha llegado el momento de empezar nuestro libro/trabajo/artículo/carta o lo que a cada uno le dicte la inspiración. Como estamos bajo el influjo de las musas, no es el momento de fijarnos en detalles de formato: unas veces empezamos el pár-

rafo con tabuladores y otras no, algunas palabras están separadas con más de un espacio, algunos párrafos están separados con dos saltos de línea, etc. Como se observa en la Figura 3, el resultado no es muy estético, así que luego habrá que pagarle a un tipógrafo para que haga la composición... o podemos usar Latex. Vamos a introducir el texto en la “plantilla” del apartado anterior, aunque le añadiremos algunos detalles, quedando como el Listado 1.

Sólo queda ejecutar latex y ver el resultado (ver la Figura 4). Impresionante, ¿verdad? Aparte de corregir el espaciado, la tabulación y la justificación, uno puede preguntarse de dónde ha salido la palabra “Capítulo” o la fecha actual en la página de título, o quién ha definido los márgenes. Todos esos detalles de formato están a cargo de Latex, que los resuelve a la perfección, aunque también es verdad que sin pre-

guntarnos nuestra opinión. Para aprender a hacer nuestros propios documentos, cada vez más perfeccionados, estudiaremos en detalle nuestro “programa” Latex.

En primer lugar, la sintaxis de Latex es bastante simple: todas las instrucciones empiezan con un backslash (la barra hacia atrás `\`), seguidas del nombre de la instrucción. La mayoría de las instrucciones requieren especificar uno o más parámetros, aunque para algunas basta escribir su nombre, por ejemplo `\maketitle`. Los parámetros opcionales, que se pueden omitir, van entre corchetes tras el nombre de la instrucción, mientras que los parámetros que son obligatorios en una instrucción se escriben entre llaves, al final de la instrucción. Hay básicamente dos clases de instrucciones: los mandatos simples y los entornos. Estos últimos son los que se definen con una pareja de instrucciones `\begin ... \end`.

En el documento anterior, sólo hay un entorno: el entorno `document`, que debe existir en todo documento Latex. Naturalmente, sólo puede haber un entorno `document`, pero si alguien está preparado para afrontar la gestión de errores de Latex, puede probar qué pasa cuando se incluye más de uno. Lo que hay dentro del entorno `document` es propiamente el contenido de nuestro documento, como su propio nombre indica, mientras que lo anterior

Tabla 1. Algunas opciones de la instrucción `\documentclass`

Opciones	Significado	Opción por defecto
10pt / 11pt / 12 pt	Tamaño de letra	10 pt
oneside / twoside	Impresión a doble cara	twoside en book, oneseide en report y article
openright / openany	Empezar capítulos en	openright en book, openany página impar
en report		
onecolumn / twocolumn	Texto en una o dos columnas	onecolumn
titlepage / notitlepage	Salto de página tras los títulos	titlepage en book y report, notitlepage en article

a `\begin{document}` es el preámbulo, donde se definen algunas características generales del documento. Como ya se habrá adivinado, se pueden introducir comentarios en cualquier punto, comenzando la línea con el símbolo `%`, de forma que estas líneas no afectan en nada al documento impreso.

El preámbulo del documento

La primera instrucción del documento es la instrucción `\documentclass`:

```
\documentclass[a4paper]{report}
```

Esto define la “clase” del documento. En LaTeX hay predefinidas tres clases de uso general: *book*, *report* y *article*. Existen también las clases *letter* (para cartas, con una gestión muy potente de etiquetas para hacer envíos masivos de cartas personalizadas) y *slides* (¡se pueden hacer presentaciones con LaTeX!) pero tienen muchas instrucciones específicas y requieren un estudio especial. No hay grandes diferencias entre *book*, *report* y *article*, más que en cuestiones de formato. En principio, la clase *book* está pensada para libros o documentos muy largos, la clase *article* se orienta a artículos de revistas, más cortos y con la clase *report*, que estaría en un punto intermedio entre las otras dos, se hacen informes técnicos. Seguramente, para hacer un trabajo escolar, la clase más adecuada sea *report*, de todas formas, lo más recomendable es probar con las tres para decidir cuál se adapta más a las características de nuestro trabajo. Si, por

ejemplo, cambiamos la clase a *book* en nuestro documento, el único cambio aparente es la introducción de una página en blanco después del título. ¿Por qué aparece esta página? Porque, en un libro, todos los capítulos empiezan en una página impar, es decir, en la página de la derecha, tal como la vemos si mantenemos el libro abierto. Para que esto sirva para algo, tenemos que imprimir el documento a doble cara. En cambio, si cambiamos a la clase *article*, se elimina la página de portada con el título. Además, se pierde el formato del título del capítulo porque, en realidad, un artículo no puede tener capítulos, sino “secciones”. Volveremos a esta cuestión pronto, al estudiar la estructura del documento.

Habiendo seleccionado la clase de nuestro documento, se pueden configurar diferentes variantes, dando parámetros opcionales en la instrucción `\documentclass`. Por ejemplo, en nuestro documento hemos especificado que el tamaño de papel es A4, con la opción *a4paper*. Podíamos también haber elegido *a5paper* (14,8 x 21), *letterpaper* (la opción por defecto: 8,5 x 11 pulgadas) o incluso otros más exóticos, pero es poco probable que se necesiten. Incluyo otras opciones, cuyo significado es autoexplicativo, en la Tabla 1. Obsérvese que con estas opciones se puede cambiar el comportamiento por defecto de una clase. Por ejemplo, se puede usar la clase *book* sin página de títulos (*notitlepage*), o la clase *report* empezando los capítulos en página impar (*openright*). Eso no quiere decir que todas las combinaciones tengan sentido, por ejemplo, es absurdo usar *openright* u *openany* en la clase *article*, que no tiene capítulos. Lo mejor es probar hasta dar con el aspecto deseado. Especialmente recomiendo la opción *twocolumn*: con una sola palabra, obtenemos el diseño en dos columnas típico de periódicos o revistas. Por cierto, al hacer todas estas pruebas, notaréis una de las principales ventajas de la mayoría de los programas visualizadores de ficheros DVI: el propio programa detecta cuándo tiene

que recargar el documento DVI porque se ha recompilado el fuente LaTeX.

Las otras dos líneas del preámbulo de nuestro documento son instrucciones `\usepackage`, que cargan paquetes adicionales. La posibilidad de extensión de LaTeX con paquetes externos es lo que le da una enorme potencia a LaTeX. Para cualquier cosa que queramos hacer, seguro que alguien ha hecho ya el correspondiente paquete y sólo tendremos que encontrarlo. Incluso podemos encontrar clases enteras para propósitos específicos, distintas de *book*, *report* y *article*. Además, hay algunos paquetes tan útiles que se han convertido en parte de la distribución estándar de LaTeX. Veamos cuáles son los dos paquetes que hemos incluido en nuestro documento, empezando por *babel*. Es gracias a la línea...

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

...como LaTeX ha sabido escribir “Capítulo”. Para aprender idiomas, basta cambiar *spanish* por *german* o *serbian* y observar la fecha y el título del capítulo. Los códigos de las lenguas oficiales de España, además de *spanish*, son *basque*, *catalan* y *galician*. ¿Adivináis cuál es el idioma por defecto si no se carga *babel*? Por cierto, al hacer cambios de idioma y recompilar, puede que se nos produzca un error. Se debe a que LaTeX guardó algunas características del documento en un fichero auxiliar, que ahora ya no resulta coherente con el documento. Para evitar el error y garantizar que partimos de cero en cada compilación, basta con borrar este fichero que, en nuestro caso, se llamaría *libro.aux*.

El otro paquete que hemos cargado en el preámbulo es *inputenc*, al escribir la línea:

```
\usepackage[latin1]{inputenc}
```

La utilidad de este paquete es tanta, que casi no se puede explicar con palabras:

RECURSOS

- [1] Comprehensive Tex Archive Network: <http://www.ctan.org> [2] *Mirror español de CTAN*: <ftp://ftp.rediris.es/sites/ftp.dante.de/tex-archive/systems/linux> [3] *Ayuda de LaTeX dentro de CTAN*: <http://www.ctan.org/tex-archive/info>

Tabla 2. Algunas instrucciones y entornos usuales en LaTeX.

Instrucción	Uso
<code>\begin{center}</code> ... <code>\end{center}</code>	Justificación centrada (se puede cambiar <i>center</i> por <i>flushright</i> -derecha- o <i>flushleft</i> -izquierda-)
<code>\begin{enumerate}</code> <code>\item</code> <code>\item</code> ... <code>\end{enumerate}</code>	Relación numerada (se puede cambiar <i>enumerate</i> por <i>itemize</i> -viñetas-) ...
<code>\footnote{Nota}</code>	Nota al pie
<code>\label{etiqueta}</code>	Marca una posición para referenciarla
<code>\ref{etiqueta}</code>	Referencia a una posición marcada
<code>\$... \$</code>	Fórmulas matemáticas en el texto
<code>\begin{equation}</code> ... <code>\end{equation}</code>	Fórmulas matemáticas en una línea aparte, con número de ecuación
<code>_ {texto}</code>	Texto en subíndice. ¡ Sólo en fórmulas !
<code>^ {texto}</code>	Texto en superíndice. ¡ Sólo en fórmulas !

basta con quitar esta línea (o ponerle un % al principio) y ver qué pasa. En los tiempos de los terminales alfanuméricos, había que teclear combinaciones como `{\e}` para escribir una letra ‘e’ con acento. Afortunadamente, esto es prehistoria tecnológica y teniendo el teclado bien configurado podemos escribir normalmente gracias al paquete *inputenc* que, por cierto, significa *INPUT ENCoding*. Se pueden usar otras codificaciones de caracteres en lugar de *latin1*, como *ansinew*, pero, al menos para textos en español o inglés, la diferencia es inapreciable.

La página de títulos

El entorno *document* empieza con los títulos, que se definen con las instrucciones `\title`, `\author` y, opcionalmente, `\date`. Cada uno de estos comandos toma como argumento un texto, que Latex imprimirá con el tipo y tamaño de letra adecuados. Este texto puede ser vacío. Por ejemplo, al imprimir la versión final de un libro no es habitual poner la fecha, así que escribiríamos `\date{}`. Una vez definidos estos campos, la instrucción `\maketitle` los muestra. Lo que debemos entender de este proceso es que el formato de la página de títulos está codificado al definir la instrucción `\maketitle` en la clase y este formato no se puede modificar, al menos no fácilmente. Por ejemplo, si queremos incluir un subtítulo, a lo mejor se nos ocurre que bastaría con insertar insertar una línea de texto entre `\title` y `\author`. Pero si pasamos el texto por procesador *latex*, descubrimos que esto no va a funcionar, porque Latex imprimirá primero el texto que está fuera de las instrucciones y luego `\maketitle` formateará la página de títulos como siempre. En cambio, tenemos bastante libertad al definir los textos de título, autor y fecha, por ejemplo, tras el `begin{document}` podemos escribir:

```
\title{El maestro de los
collares
\l[0.5 cm]{\small Tres
collares para los príncipes
feos sobre la tierra
```

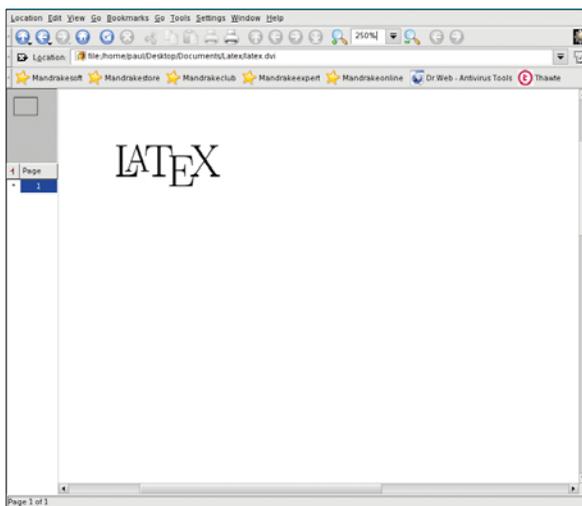


Figura 5: El símbolo de LaTeX se genera con... LaTeX

```
\ Siete para las señoras
grandotas en castillos de
mármol}}
\author{J.R. Talco \
Traducción: Un servidor}
\date{2004 \
\textsc{Málaga}}
\maketitle
```

La instrucción `\l` incluye una línea en blanco y, si le sigue una longitud entre corchetes, añade un espacio vertical adicional. En general, es preferible evitar este tipo de cosas en el texto, porque Latex ya pone el espaciado adecuado, pero el título es un caso particular. La instrucción `\textsc` escribe un texto en un tipo de letra especial (“Small Capitals”), como `\textsf` (“Sans serif”), `\textsl` (“Slanted”) o `\texttt` (“Typewriter”). El texto en negrita se incluye con `\textbf` (“Boldface”), mientras que, para la cursiva es conveniente utilizar `\emph` (“Emphasize”), aunque se puede poner también `\textit` (“Itálica”).

El cuerpo del documento

El documento propiamente dicho es lo más fácil, ya que todo el formato está ya definido, por lo que sólo hay que escribir el texto. Si se quiere, se pueden cambiar los tipos de letra con las instrucciones que comentamos en el párrafo anterior. Entre dos párrafos hay que dejar una línea en blanco. Para comenzar un nuevo capítulo, basta dar una nueva instrucción `\chapter` y, además dentro de cada capítulo, es posible ir definiendo estructuras cada vez más pequeñas: `\section`, `\subsection` y `\subsubsection`.

Naturalmente, Latex controla la numeración de las secciones a la perfección: si queremos cambiar de orden dos capítulos basta con cortar y pegar sus contenidos y, mientras que en un procesador de textos tendríamos que revisar la numeración, con LaTeX, los capítulos se numeran automáticamente. Basta con escribir varias páginas para darnos cuenta de todo lo que hace automáticamente este maravilloso, por ejemplo, las cabeceras de página incluyen el número de página y el título del capítulo actual y se distinguen la página derecha de la izquierda, de forma que el libro quede perfecto al encuadernar. Y, cuando llevemos escritas 500 páginas, se nos ocurre que queremos un índice al principio: no hay problema, basta un `\tableofcontents` después del `\maketitle`. Eso sí, posiblemente haya que ejecutar *latex* dos o tres veces seguidas para que genere los ficheros auxiliares que necesita para construir el índice, pero pronto apreciaremos la potencia de LaTeX y como la curva de aprendizaje necesario para dominarlo se compensa con las funcionalidades que proporciona.

Llegado este punto, sólo queda utilizar Latex y probar diferentes cosas. En la Tabla 2 hemos seleccionado las instrucciones de Latex que se corresponden con la barra de herramientas estándar de un procesador de textos. ¡Probad a meter unos entornos *enumerate* o *itemize* dentro de otros! Desde luego, esto es tan sólo una primera aproximación a Latex, centrada en lo que se puede hacer con el mínimo esfuerzo. Para explicar las fórmulas matemáticas, las tablas y las figuras, o cómo modificar el formato de página, se requiere más espacio y, de hecho, es un tema que abordaremos el mes que viene, aquí, en nuestro curso de LaTeX. ■

EL AUTOR

Hace ya más de una década que Miguel Atencia viene administrando sistemas y redes en distintos organismos. Posteriormente, estudió matemáticas y consagró su vida a la enseñanza y a la ciencia. No obstante, sigue en contacto con el mundo de la informática, pues no ha podido aún librarse de desatascar impresoras y formatear ordenadores.