

# L<sup>A</sup>T<sub>E</sub>X para usuarios de procesadores de texto

Antoni Aloy López

---

BULMA

---

21 de septiembre de 2003

## **Resumen**

$\text{\LaTeX}$  no está limitado a la escritura de textos científicos y técnicos, sinó que es un excelente substituto a los procesadores de textos tradicionales en apartados en los que éstos se quedan cortos. Este manual pretende arrojar un poco de luz sobre las características de  $\text{\LaTeX}$  que lo hacen ideal para la escritura de textos extensos sustituyendo a los típicos procesadores gráficos WYSIWYG.

Las posibilidades de  $\text{\LaTeX}$  son inmensas, las combinaciones y ordenes que se pueden utilizar, o crear en el caso de que no existan tienen a infinito. En este documento veremos sólo una ínfima parte, las más comunes, y aún siendo pocas nos permitirán escribir documentos de gran calidad, concentrándonos en el contenido y no en la forma.

# Índice general

<b>1. <math>\text{\LaTeX}</math>?</b>	<b>3</b>
1.1. Introducción . . . . .	3
1.2. ¿Cuándo optar por $\text{\LaTeX}$ ? . . . . .	3
1.3. Editores $\text{\LaTeX}$ . . . . .	4
1.3.1. ¿Cuál elegir? . . . . .	4
1.3.2. ¿Y si tengo que utilizar Hase? . . . . .	5
<b>2. Utilizar <math>\text{\LaTeX}</math></b>	<b>6</b>
2.1. Nuevo documento . . . . .	6
2.2. Empezando a escribir . . . . .	7
2.2.1. Carácteres especiales . . . . .	8
2.2.2. Estructurar el documento . . . . .	8
<b>3. Documentos profesionales</b>	<b>9</b>
3.1. Formato . . . . .	9
3.1.1. Párrafo . . . . .	9
3.1.2. Carácter . . . . .	13
3.1.3. Página . . . . .	14
3.2. Tablas . . . . .	16
<b>4. A toda máquina!</b>	<b>18</b>
4.1. Documentos largos . . . . .	18
4.2. Imágenes . . . . .	18
4.3. Líneas . . . . .	19
4.4. Algunos paquetes interesantes . . . . .	19
4.5. Fórmulas matemáticas . . . . .	21
<b>5. Otras fuentes de información</b>	<b>24</b>
5.1. Enlaces . . . . .	24
5.2. Bibliografía . . . . .	24

# Índice de cuadros

2.1. Carácteres especiales . . . . .	8
3.1. Formatos de carácter . . . . .	13
3.2. Tamaño de carácter . . . . .	14
3.3. Fuentes disponibles . . . . .	14
3.4. Tabla de ejemplo . . . . .	17
4.1. <code>bbding</code> . . . . .	22
4.2. Fórmulas básicas . . . . .	23

# Capítulo 1

## L<sup>A</sup>T<sub>E</sub>X?

### 1.1. Introducción

Este artículo está pensado para aquellas personas que vienen del mundo de los procesadores de textos clásicos, normalmente procesadores gráficos de tipo WYSIWYG. La aproximación a L<sup>A</sup>T<sub>E</sub>X se hará desde el punto de vista de los trabajos más habituales que se abordan en este tipo de procesadores y veremos las soluciones que nos ofrece L<sup>A</sup>T<sub>E</sub>X para las tareas más habituales.

Normalmente se introduce L<sup>A</sup>T<sub>E</sub>X como la solución a la hora de escribir documentos con una fuerte carga matemática. Aquí la aproximación va a ser radicalmente distinta. En nuestro caso L<sup>A</sup>T<sub>E</sub>X es la herramienta que nos va a permitir la escritura de documentos largos, con una estructura muy bien definida y en los que queramos incidir más en el contenido que no en cómo se presenta la información. La presentación se la dejaremos a un experto L<sup>A</sup>T<sub>E</sub>X que se asegurará que nuestro documento tiene un acabado profesional.

Este artículo está basado en L<sup>A</sup>T<sub>E</sub>X for Word Processor Users de Guido Gonzato y en mi experiencia personal en la utilización de L<sup>A</sup>T<sub>E</sub>X para la escritura de manuales técnicos e informes. Con esto quiero decir que no esperéis una traducción literal ni siquiera aproximada del documento de Guido, lo que sí he intentado hacer es seguir el espíritu del documento y escribir un pequeño manual que anime a la gente a probar L<sup>A</sup>T<sub>E</sub>X y no lo voy a negar, realizado algún que otro copiar y pegar para ahorrarme el trabajo de escribir algún que otro ejemplo. La guía de Guido además de ser una fuente de información y de referencia es un ejemplo claro de lo que se puede hacer con L<sup>A</sup>T<sub>E</sub>X y vale la pena estudiar el código fuente del documento.

### 1.2. ¿Cuándo optar por L<sup>A</sup>T<sub>E</sub>X ?

Mi experiencia personal con L<sup>A</sup>T<sub>E</sub>X es mas bien curiosa. Cuando tenía necesidad de escribir fórmulas matemáticas complejas no tenía acceso a la herramienta, y cuando lo he tenido lo he utilizado no para este trabajo, sino como solución a un problema que los procesadores de textos clásicos manejaban muy mal: el trabajo con textos de varios cientos de páginas.

Si llegar al extremo de utilizar L<sup>A</sup>T<sub>E</sub>X para escribir cualquier tipo de documento, que de poder, se puede, yo me atrevería a sugerir la utilización de esta excelente herramienta principalmente para los siguientes casos:

- En documentos con muchos elementos matemáticos
- En documentos muy extensos y bien estructurados
- Cuando la portabilidad entre plataformas sea importante
- Cuando esté prevista la generación del mismo documento en diversos formatos.

Las herramientas y librerías que nos brinda  $\text{\LaTeX}$  son legión y podemos utilizarlo tanto para maquetar nuestra partida de ajedrez como para crear en un plis plas una presentación, escribir un poema o redactar la carta a la novia, el límite es nuestra imaginación y nuestras ganas de experimentar con la herramienta.

### 1.3. Editores $\text{\LaTeX}$

Un documento  $\text{\LaTeX}$  es texto puro y duro. Puede utilizarse, por tanto cualquier procesador de textos capaz de guardar el documento en texto plano. Veamos algunos ejemplos:

- Vi, Vim, Emacs, jed, ... Son los clásicos. Entorno consola y con resaltado de sintaxis para  $\text{\LaTeX}$ , lo que ayuda mucho a la hora de detectar errores.
- LyX i KLyX, son procesadores de texto casi-WYSIWYG, que nos puede ayudar a introducirnos en el mundillo  $\text{\LaTeX}$
- GNU  $\text{\TeX}$ MACS un potente procesador WYSIWYG, muy potente aunque con algunos problemas de lentitud que se van corrigiendo versión a versión.
- Kile. Mi favorito. Un entorno integrado de edición para  $\text{\LaTeX}$ , muy parecido a un IDE de programación. Es básicamente un editor de texto plano junto con un conjunto de botones de ayuda con las principales funciones de  $\text{\LaTeX}$  y accesos rápidos a la compilación del documento y visualización del resultado.

#### 1.3.1. ¿Cuál elegir?

Si disponemos de un entorno gráfico yo me inclino por Kile, básicamente porque presenta la mejor relación entre potencia, facilidad de uso y un nivel de "intrusismo" mínimo, que me permite concentrarme en lo que estoy escribiendo.

La pega de estos editores de texto plano es que al principio choca bastante el ver códigos de formateo de texto entre el escrito, pero uno se acostumbra a todo, así donde vosotros veis un simple código

```
\textbf{texto}
```

yo veo a una rubia, digo a una negrita, digo un texto en negrita, ya me entendéis.

### 1.3.2. ¿Y si tengo que utilizar Hase?

No problem!  $\text{\LaTeX}$  está portando también a este "sistema operativo". Buscad en la web  $\text{MiKTeX}$  y el editor de textos recomendado el  $\text{TeXnicCenter}$ . Esto os permitirá editar el documento en ambos sistemas.

En Linux las herramientas de que disponemos son más y mejores, y la compilación más rápida, pero tanto  $\text{MiKTeX}$  como  $\text{TeXnicCenter}$  nos pueden sacar de un apuro. Además tanto  $\text{MiKTeX}$  como  $\text{TeXnicCenter}$  son proyectos de código abierto, por lo que también podemos aprovechar para introducir esta filosofía en los endornos Hasefroch.

# Capítulo 2

## Utilizar L<sup>A</sup>T<sub>E</sub>X

### 2.1. Nuevo documento

El equivalente a un nuevo documento el L<sup>A</sup>T<sub>E</sub>X sería

```
\documentclass[a4paper,12pt]{article}
\usepackage[latin1]{inputenc}
\usepackage[spanish]{babel}
\begin{document}
%escribir aquí
\end{document}
```

Con esto tenemos el equivalente a una página en blanco, preparada para escribir un artículo en un papel A4, con un cuerpo de letra de 12 puntos y con el guionado y las fuentes preparadas para el idioma español.

Aprovecharé esta pequeña plantilla para introducir el concepto de librería o paquete en L<sup>A</sup>T<sub>E</sub>X. Mediante la orden *usepackage* le indicamos a L<sup>A</sup>T<sub>E</sub>X que cargue un conjunto de macros que implementarán una funcionalidad, en nuestro caso la codificación y el particionado.

Este sistema, aunque implica conocer y buscar la librería que queramos utilizar nos permite expandir enormemente las posibilidades de L<sup>A</sup>T<sub>E</sub>X.

Por ejemplo, si queremos dar una mejor imagen a nuestro documento, añadiéndole cabeceras y pies de página, cargaremos el paquete *fancyhdr*. Si queremos que L<sup>A</sup>T<sub>E</sub>X sea capaz de interpretar correctamente el símbolo del Euro, añadiremos *marvosym*.

Una plantilla más elegante para nuestros documentos sería por ejemplo esta

```
\documentclass[a4paper,12pt]{article}
\usepackage{times}
\usepackage[latin1]{inputenc}
\usepackage[spanish]{babel}
\usepackage{fancyhdr}
\usepackage{marvosym}
\usepackage{hyperref}

\pagestyle{fancy}
% Aquí ponemos nuestra cabecera
```



```

\lhead{}
\chead{}
\rhead{\bfseries Bulma}
\lfoot{\LaTeX{}}
\rfoot{aaloy}
\cfoot{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}

\begin{document}

% Aquí el título
\title{{\Huge \LaTeX{}} }\\
\author{aaloy} }
\maketitle
\newpage
% Y el índice
\tableofcontents
\newpage

% Y a empezar a escribir
\section {Introducción}

\end{document}

```

Esta plantilla tiene la estructura más básica de un artículo, con página de presentación, índice y una sección, a la vez que le añade la elegancia de la cabecera y pie de página. Es la estructura que se ha empleado en la redacción de este documento y que me sirve a mi de plantilla para mis documentos L<sup>A</sup>T<sub>E</sub>X.

## 2.2. Empezando a escribir

Antes de empezar a escribir conviene pensar antes en la estructura que tendrá el documento. Es conveniente acostumbrarse a pensar en términos de partes, capítulos, secciones, subsecciones, subsubsecciones,... Algunas veces lo que yo suelo hacer es empezar creando la estructura básica del documento, normalmente capítulos, secciones y alguna subsección y a partir de ahí voy rellenando.

También nos resultará chocante al principio la manera en que L<sup>A</sup>T<sub>E</sub>X trata las líneas en blanco: *simplemente actúan como separadores de párrafo*. y son equivalentes a poner `\\` detrás de una línea. Del mismo modo son aunque nos empeñemos en separar las palabras por múltiples espacios en blanco, L<sup>A</sup>T<sub>E</sub>X los trata como si fueran uno solo, así

Esto                                  es    una                                  prueba

Al pasar por el filtro de L<sup>A</sup>T<sub>E</sub>X acaba como

Esto es una prueba

Lo que aparentemente puede parecer un problema se transforma en una virtud, podemos dar claridad a nuestro escrito empleando líneas en blanco y espacios y dejar que L<sup>A</sup>T<sub>E</sub>X haga el trabajo duro de la maquetación final.

### 2.2.1. Carácteres especiales

Otro de los problemas más frecuentes con los que nos encontraremos es el del uso de caracteres especiales de L<sup>A</sup>T<sub>E</sub>X en nuestro texto. Al tener un significado especial en L<sup>A</sup>T<sub>E</sub>X producirán efectos indeseados que van desde el no aparecer en pantalla a no dejarnos compilar nuestro documento. En la Tabla 2.1 están los más usuales.

Carácter	Debes escribir	Carácter	Debes escribir
\$	<code>\\$</code> or <code>\textdollar</code>	&	<code>\&amp;</code>
%	<code>\%</code>	-	<code>\_</code> or <code>\textunderscore</code>
{	<code>\{</code> or <code>\textbraceleft</code>	}	<code>\}</code> or <code>\textbraceright</code>
<	<code>\$&lt;\$</code> or <code>\textless</code>	>	<code>\$&gt;\$</code> or <code>\textgreater</code>
\	<code>\textbackslash</code>		<code>\textbar</code>
•	<code>\textbullet</code>	‡	<code>\textdaggerdbl</code>
†	<code>\textdagger</code>	¶	<code>\textparagraph</code>
§	<code>\textsection</code>	©	<code>\textcopyright</code>
^	<code>\textasciicircum</code>	~	<code>\textasciitilde</code>
~	<code>\$\sim\$</code>	®	<code>\textregistered</code>
™	<code>\texttrademark</code>	<sup>a</sup>	<code>\textordfeminine</code>
°	<code>\textordmasculine</code>		

Cuadro 2.1: Carácteres especiales

### 2.2.2. Estructurar el documento

Para estructurar nuestro documento utilizaremos la estructura `\[nivel]{nombre}`, donde de mayor menor nivel tendremos:

- `part`
- `chapter`
- `section`
- `subsection`
- `subsubsection`

Si añadimos un `*` delante la primera llave antes del nombre no se numerará la el nivel. Por ejemplo, si no queremos numerar un nivel de tipo `section` haríamos

```
\section *{Esto no va numerado}
```

# Capítulo 3

## Documentos profesionales

### 3.1. Formato

#### 3.1.1. Párrafo

Por defecto  $\text{\LaTeX}$  tiene activada la justificación completa de párrafos. Podemos cambiar este tipo de justificación según nos interese utilizando las órdenes *flushleft*, *center* y *flushright*, para lograr que el párrafo nos quede alineado a la izquierda, centrado o alineado a la derecha. Veámoslo con unos ejemplos:

Esto es un párrafo alineado a la izquierda. Esto quiere decir que no se mantiene la justificación completa propia de  $\text{\LaTeX}$ .  
Se logra con

```
\begin{flushleft}  
(aquí escribe tu texto)  
\end{flushleft}
```

Esto es un párrafo centrado. Esto quiere decir que no se mantiene la justificación completa propia de  $\text{\LaTeX}$ .  
Se logra con

```
\begin{center}  
(escribe aquí tu texto)  
\end{center}
```

Esto es un párrafo alineado a la derecha. Esto quiere decir que no se mantiene la justificación completa propia de  $\text{\LaTeX}$ .  
Se logra con

```
\begin{flushright}  
(escribe aquí tu texto)  
\end{flushright}
```

## Creación de párrafos

Recordemos que para forzar un cambio de línea en  $\text{\LaTeX}$  se utiliza la doble diagonal invertida  $\backslash$  ya que de lo contrario  $\text{\LaTeX}$  lo que hace es eliminar los espacios sobrantes.

Cuando nos interese iniciar un nuevo párrafo podemos hacerlo dejando una o más líneas en blanco o mediante la orden  $\backslash par$ . Esta orden que salte una línea y que comience en un párrafo nuevo.

Podemos controlar el espaciado entre dos párrafos utilizando  $\backslash[espacio]$  donde *espacio* indica la medida que podemos dejar, por ejemplo:

$\backslash[15pt]$  deja un espaciado equivalente a 15 puntos.

$\backslash[.5cm]$  dejaría un espaciado de 0.5 cm.

## Interlineado

Podemos cambiar el interlineado de un párrafo haciendo uso del paquete `setspace`, que define los entornos *singlespace*, *onehalfspace*, *doublespace*, para el interlineado normal, línea y media y doble línea respectivamente.

Este paquete también nos permite utilizar un interlineado *ad-hoc* mediante el uso de `spacin{num líneas}`.

```
\begin{doublespace}
```

Esto tiene un interlineado  $\backslash$  a doble línea  $\backslash$  usando `setspace`

```
\end{doublespace}
```

Esto tiene un interlineado

a doble línea

usando `setspace`

```
\begin{spacing}{3}
```

Esto es un interlineado  $\backslash$  de tres líneas

```
\end{spacing}
```

```
\begin{spacing}{3}
```

Esto es un interlineado  $\backslash$

de tres líneas

```
\end{spacing}
```

Esto es un interlineado

de tres líneas

```
\begin{spacing}{0.5}
```

```
\tiny
```

Esto es un interlineado  $\backslash$

de media línea

```
\end{spacing}
```

Esto es un interlineado  
de media línea

## Numeración y viñetas

El mundo se divide en dos grandes clases:

- Aquellos que hacen listas
- Aquellos que no

L<sup>A</sup>T<sub>E</sub>X nos proporciona potentes herramientas de confección de listas, utilizando para ello tres tipos de entornos: *itemize*, *enumerate* y *description*. El primer entorno nos proporciona una lista no numerada, el segundo una lista numerada y el tercer entornos nos crea un tipo de lista apto para las definiciones.

Las listas se pueden anidar, y L<sup>A</sup>T<sub>E</sub>X se encarga de que el indentado y la numeración (en el caso de una lista enumerada), por lo que sólo tendremos que preocuparnos de escribir.

El ejemplo clásico

```
\begin{itemize}
\item uno
\item dos
\item tres
\end{itemize}
```

- uno
- dos
- tres

El mismo ejemplo pero cambiando una de las viñetas. Fijémonos que únicamente tenemos que indicar entre corchetes el símbolo que queremos utilizar.

```
\begin{itemize}
\item[*] uno
\item dos
\item tres
\end{itemize}
```

- \* uno
- dos
- tres

Una lista con las viñetas personalizadas. Fijémonos cómo se utiliza el comando `\renewcommand` para indicar la viñeta que queremos en cada nivel.

```
\begin{itemize}
\renewcommand{\labelitemi}{*}
\renewcommand{\labelitemii}{-}
\item nivel 1, item 1
\item nivel 1, item 2
\begin{itemize}
\item nivel 2, item 1
\item nivel 2, item 2
\end{itemize}
\item nivel 1, item 3
\end{itemize}
```

- \* nivel 1, item 1
- \* nivel 1, item 2
  - nivel 2, item 1
  - nivel 2, item 2
- \* nivel 1, item 3

Ahora le toca el turno a las listas numeradas.

```
\begin{enumerate}
\item un
\item dos
\item tres
\end{enumerate}
```

1. un
2. dos
3. tres

```
\begin{enumerate}
\item un
\begin{enumerate}
\item Primer subapartado
\item Segundo subapartado
\end{enumerate}
\item dos
\item tres
\end{enumerate}
```

- |    |   |
|----|---|
| 1. | un  |
|    | <ul style="list-style-type: none"> <li>a) Primer subapartado</li> <li>b) Segundo subapartado</li> </ul> |
| 2. | dos   |
| 3. | tres  |

Las listas numeradas nos permiten hasta **cinco** niveles. Podemos redefinir el tipo de numeración utilizada en cada uno de los subniveles mediante las órdenes `labelitemi`, `labelitemii`, `labelitemiii`, `labelitemiv` y utilizar en lugar de la numeración estandar (`\arabic`) números romanos (`\roman` o bien `\Roman`) o letras (`\alph` o bien `\Alph`).

Podemos combinar varios tipos de numeraciones, así por ejemplo escribiendo

```
\begin{enumerate}
\renewcommand{\labelenumi}{\Roman{enumi}}
\renewcommand{\labelenumii}{\alph{enumii}}
\renewcommand{\labelenumiii}{\arabic{enumiii}}
\item Nivel superior
\item En números romanos
\begin{enumerate}
\item Segundo nivel
\item Segundo nivel con opciones
\begin{enumerate}
\item Tercer nivel
\item Y este también
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

obtenemos la siguiente lista:

I Nivel superior

II En números romanos

a Segundo nivel

b Segundo nivel con opciones

1 Tercer nivel

2 Y este también

El entorno *description* es el adecuado para realizar definiciones. Entre corchetes situaremos el elemento a definir, que actúa en este caso como las viñetas de los entornos anteriores.

```
\begin{description}
\item[uno] uno
\item[dos] dos
\end{description}
```

<b>uno</b> uno
<b>dos</b> dos

L<sup>A</sup>T<sub>E</sub>X no nos limita a las viñetas clásicas. Por ejemplo el paquete *pifont* nos proporciona una gran cantidad de símbolos que podemos utilizar y algunas órdenes especialmente adaptadas a conseguir vistosas listas:

```
\begin{dinglist}{43}
\item uno
\item dos
\item tres
\end{dinglist}
```

☞ uno  
☞ dos  
☞ tres

```
\begin{dingautolist}{172}
\item uno
\item dos
\item tres
\end{dingautolist}
```

① uno  
② dos  
③ tres

### Indentación

Podemos suprimir la indentación de un párrafo con la instrucción `\noindent`. Si por el contrario lo que queremos es establecer la longitud de la indentación de un párrafo escribiremos por ejemplo `\setlength{\parindent}{2 cm}` que nos indentaría el párrafo 2 cm.

### 3.1.2. Carácter

En nuestros escritos es muchas veces conveniente realzar una parte bien con negritas, cursiva, cambiando el tipo de letra o su tamaño. Veremos como podemos hacerlo en L<sup>A</sup>T<sub>E</sub>X

Para el estilo	Debes escribir
<i>texto enfatizado</i>	<code>\emph{texto enfatizado}</code>
<b>texto en negrita</b>	<code>\textbf{texto en negrita}</code>
<i>texto en cursiva</i>	<code>\textit{texto en cursiva}</code>
<b><i>negrita cursiva</i></b>	<code>\textbf{\textit{negrita cursiva}}</code>
texto normal	<code>\textnormal{texto normal}</code>
VERSALITA	<code>\textsc{Versalita}</code>

Cuadro 3.1: Formatos de carácter

Estos atributos son combinables entre sí y dentro de un mismo párrafo, por lo que es posible escribir en un mismo párrafo multitud de combinaciones.

```
\huge{Esto es} una \textbf{\tiny{negrita pequeña}}
```

Esto es una negrita pequeña

Para el tamaño	Debes escribir
<b>Huge</b>	<code>\Huge{texto}</code>
<b>huge</b>	<code>\huge{texto}</code>
<b>LARGE</b>	<code>\LARGE {texto}</code>
<b>Large</b>	<code>\Large {texto}</code>
<b>large</b>	<code>\large {texto}</code>
<code>normalsize</code>	<code>\normalsize {texto}</code>
<code>small</code>	<code>\small {texto}</code>
<code>footnotesize</code>	<code>\footnotesize {texto}</code>
<code>scriptsize</code>	<code>\scriptsize {texto}</code>
<code>tiny</code>	<code>\tiny {texto}</code>

Cuadro 3.2: Tamaño de carácter

Para la familia de fuente	Debes escribir
<code>roman</code>	<code>\roman{texto}</code>
<b>sanserif</b>	<code>\sanserif{texto}</code>
<code>ttfamily</code>	<code>\ttfamily{texto}</code>

Cuadro 3.3: Fuentes disponibles

### El símbolo del Euro

El paquete *marvosym* incluye varios caracteres interesantes, entre ellos el símbolo del Euro €. Lo podemos obtener tecleando `\EUR`.

### 3.1.3. Página

Normalmente  $\text{\LaTeX}$  nos controlará perfectamente los saltos de página, pero también nos permite forzar cuando queramos un salto de página en cualquier punto del documento utilizando la orden `\newpage` o bien `\clearpage`.

### Tipo y Orientación

Los tipos básicos de página son *a4paper*, *a5paper*, *b5paper*, *letterpaper*, *legalpaper* y *executivepaper*. Normalmente la que más utilizarás será la *a4paper*.

La orientación viene puede ser horizontal *portrait* o apaisada *landscape*.

Tanto el tamaño inicial de la página como la orientación se definen como parámetros de la orden `\documentclass` que también nos define el tipo de documento que estamos escribiendo (book, report, article, letter, slides, ...).

En este documento he utilizado

```
\documentclass[12pt, titlepage, a4paper]{report}
```



## Márgenes

La manera más sencilla de establecer tamaños de márgenes, longitud de página, longitud y ancho del texto es utilizar el paquete *geometry*. Por ejemplo,

```
\usepackage[left=3cm, right=3cm]{geometry}
```

Nos establece los márgenes izquierdo y derecho de nuestro documento a 3 cm. Otras de configuración son, entre otras:

**paperwidth** Para establecer el ancho del papel

**paperheight** Establece el alto del papel

**textwidth** Para establecer el ancho del área de escritura

**textheight** Establece el alto del área de escritura

**top** Márgen superior

**bottom** Márgen inferior

## Cabecera y pie de página

Una de las maneras de tener una cabecera y pie de página más sencillas es utilizando el paquete *fancyhdr* (utilizado también en este documento).

Este paquete nos define tres zonas (izquierda, centro y derecha) en las que personalizar nuestras cabeceras y pies de página, nos permite añadir líneas decorativas y tener cabeceras y pies de página mayores que el ancho de texto que estemos utilizando para escribir el documento, entre otras muchas opciones que nada tienen que envidiar a los procesadores de texto gráficos. Para ello incluiremos en el preámbulo de nuestro documento las siguientes líneas:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

En el siguiente gráfico podemos ver la distribución básica de las distintas secciones de la cabecera y del pie de página de un documento.

Cabecera Izda	Cabecera Central	Cabecera Derecha
texto texto texto		
Pié Izdo	Pié Central	Pié Derecho

Es conveniente dedicar algo de tiempo a leer la documentación de *fancyhdr*, que es muy completa e instructiva. Pero si eres de los que prefiere aprender con el ejemplo, aquí tienes unas cuantas recetas útiles:

Por ejemplo, para conseguir:

Título del documento	
texto texto texto	
aaloy	texto central 3

escribiríamos

```
\lhead{}
\chead{}
\rhead{\bfseries Título del documento}
\lfoot{aaloy}
\cfoot{texto central}
\rfoot{Página \thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

Otra combianción simple y efectiva es esta:

```
\lfoot{\LaTeX{} para usuarios de procesadores de texto}
\cfoot{}
\rfoot{\thepage}
%\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

Fijémonos en la orden `\thepage` indica a  $\text{\LaTeX}$  que debe introducir el número de la página actual en esta posición.

Hay numerosas combinaciones y órdenes de formateo de cabeceras y piés de páginas. Una vez más os remito a la documentación del paquete. En un documento normal de texto los ejemplos presentados aquí deberían ser suficientes.

## 3.2. Tablas

Podríamos dedicar un capítulo entero a las virguerías que puede hacer  $\text{\LaTeX}$  con las tablas, pero tampoco es el objetivo de este documento, que no pasa de una simple introducción.

`\begin{tabular}{|c|l|}` nos define una table con dos columnas, con líneas de sparación verticales, donde la primera tiene el texto centrado y la segunda el texto alineado a la izquierda.

En el entorno table podemos definir una posición relativa que nos indicará cómo se situará la tabla dentro del documento

- h (here) Indica que si es posible la tabla se situará en la posición donde está el entorno table
- t (top) Indica que la tabla se situará en la parte superior de la hoja

- b (bottom) Situaría la tabla en la parte inferior de la hoja
- p (page) Indica que la tabla se situará en una hoja aparte.

La posición final de la tabla hay que tener en cuenta que la decidirá en última instancia el propio  $\text{\LaTeX}$  en función del documento que estemos escribiendo.

```
\begin{table}[htb]
\begin{center}
\begin{tabular}{|c|l|} \hline
uno & dos \\ \hline
a   & b  \\ \hline
\end{tabular}
\end{center}
\caption{Tabla de ejemplo}
\label{tab:etiqueta_referencia}
\end{table}
```

línea uno	esta es la segunda columna
a	b

Cuadro 3.4: Tabla de ejemplo

Por otra parte tenemos que tener en cuenta que el entorno *tabular* sólo permite que la tabla ocupe como máximo una hoja. Si tenemos tablas más largas, debemos optar por otros entornos, como por ejemplo *longtable*.

# Capítulo 4

## A toda máquina!

En los capítulos anteriores habíamos visto cómo  $\text{\LaTeX}$  se puede utilizar para sustituir la mayoría de funcionalidades de un procesador de texto. En este capítulo veremos cómo tratar con documentos largos, insertar imágenes y otras funciones de utilidad.

### 4.1. Documentos largos

Este documento está escrito a partir de varios archivos y luego ensamblado.  $\text{\LaTeX}$  nos permite utilizar `\include{nombre_archivo}` indicando que en ese punto de debe insertar el contenido del archivo que se indica entre las llaves.

Personalmente me gusta tener cada capítulo de un documento largo en su archivo correspondiente y luego ensamblar cada capítulo en el documento principal, que de este modo sólo incluye los paquetes utilizados, información del documento, autor, etc y varios *includes* con cada uno de los archivos.

**Atención! No se pueden anidar las inclusiones de archivos.**

```
\include{preamble}
\include{introduccion}
\include{nuevo}
\include{formato}
\include{maquina}
\include{enlaces}
```

Esto permite crear fácilmente extractos del documento que incluyan o no unos capítulos determinados. Así por ejemplo

```
\includeonly{preamble, enlaces}
```

nos crearía un documento con únicamente los índices y tablas de contenido, el contenido del archivo *preamble* y el contenido del archivo *enlaces*.

### 4.2. Imágenes

Veamos un ejemplo de cómo se incluiría la imagen *bulma.png* en el documento. Dado que se trata de una imagen un poco grande la reduciremos al 50 %.

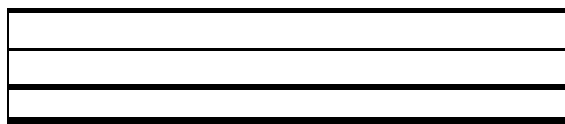
```
\begin{figure}[htbp]
\scalebox{0.5}{\includegraphics{bulma.png}}
\end{figure}
```



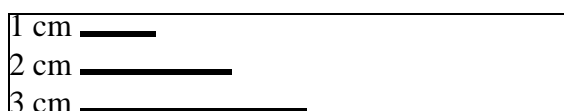
### 4.3. Líneas

Podemos dibujar líneas de cualquier longitud y grueso con la orden `\rule` indicando la longitud y el grueso de la línea que queremos dibujar.

```
\rule{\linewidth}{1pt}
\rule{\linewidth}{1pt}
\rule{\linewidth}{2pt}
\rule{\linewidth}{4pt}
```



```
1 cm \rule{1cm}{2pt}\\
2 cm \rule{2cm}{2pt}\\
3 cm \rule{3cm}{2pt}
```



### 4.4. Algunos paquetes interesantes

**schedule** Este interesante paquete tiene por objetivo facilitarnos la vida a la hora de confeccionar horarios. Para utilizarlo debemos incluir el paquete y definir cómo queremos el horario. Veamos el ejemplo que incluye en su documentación, primero define los parámetros del horario

```
\usepackage{schedule}

\CellHeight{.4in} % alto de la celda
\CellWidth{.925in} % ancho de la celda
\TimeRange{10:00-17:00} % de qué horas a qué horas va el horario
\SubUnits{30} % cada cuántos minutos hay una subdivisión
\BeginOn{Monday} % en qué día empieza la semana
\TextSize{\tiny} % Tamaño de la fuente
\FiveDay % Semana de cinco días

% definición de colores para cada tipo de evento
% <evento> <color de fondo> <color del texto>
%
\NewAppointment{class}{dark}{black}
\NewAppointment{meeting}{red}{white}
\NewAppointment{workshop}{green}{blue}
```

La introducción de un evento se hace indicando su tipo, una línea de observaciones, el día de la semana (en inglés) y el rango de horas en el que tiene lugar.

Consiguiendo el horario

Fall Quarter, 1997					
	Monday	Tuesday	Wednesday	Thursday	Friday
10:00 am					
10:30 am			Realism/Antirealism HOB2 233		
11:00 am		Math Logic EIC 128		Math Logic EIC 128	
11:30 am					
12:00 am			Departmental Meeting HOB2 233		
12:30 am					
1:00 pm	Critical Reasoning SSL 290	Crit. Reas. Workshop HOB2 233	Critical Reasoning SSL 290		Critical Reasoning SSL 290
1:30 pm					
2:00 pm	Moral Philosophy HOB2 233		Office Hours HOB2 210		Office Hours HOB2 210
2:30 pm					
3:00 pm					
3:30 pm		L&M Workshop HOB2 233			
4:00 pm					
4:30 pm					
5:00 pm					

Simplemente con el código

```
\begin{schedule}[Fall Quarter, 1997]%
  \class{Moral Philosophy}{HOB2 233}{M}{14:00-16:50}
  \class{Philosophy of Science}{HOB2 233}{M}{18:00-20:50}
  \class{Realism/Antireal\ism}{HOB2 233}{W}{10:30-12:00}
  \class{Math Logic}{EIC 128}{T,Th}{11:00-12:20}
  \class{Critical Reasoning}{SSL 290}{M,W,F}{13:00-13:50}
  \meeting{Departmental Meeting}{HOB2 233}{W}{12:00-12:50}
  \workshop{L&M Workshop}{HOB2 233}{T}{15:30-17:30}
  \workshop{Crit. Reas. Workshop}{HOB2 233}{T}{13:00-13:50}
  \class{Office Hours}{HOB2 210}{W,F}{14:00-14:50}
\end{schedule}
```

**bbding** Nos proporciona un buen conjunto de símbolos gráficos para utilizar en nuestros escritos.

**clock** Nos permite dibujar relojes en nuestro documento.

Y estos sólo son un mínimo, un ínfimo ejemplo de lo que nos podemos encontrar. La instalación standard de L<sup>A</sup>T<sub>E</sub>X ya viene con un gran número de paquetes, y en el repositorio <http://www.ctan.org> podéis encontrar un buen número más, desde paquetes para utilizar caracteres chinos, símbolos musicales, dibujar crucigramas y circuitos hasta paquetes para componer recetas de cocina.

## 4.5. Fórmulas matemáticas

Vale, ya sé que había dicho que en este documento no habría fórmulas matemáticas, pero no puedo resistir la tentación.  $\text{\LaTeX}$  es endiablidamente bueno con las fórmulas matemáticas!

Sólo veremos unos pequeños ejemplos de los métodos que tenemos para escribir fórmulas matemáticas en  $\text{\LaTeX}$ , para una información más exhaustiva de toda la simbología y de lo que se puede hacer os remito a los enlaces y la bibliografía.

El primer método es el más rápido y sencillo y sirve para fórmulas no demasiado largas y complejas. Iniciamos y finalizamos la fórmula con el signo de dólar \$. Así, por ejemplo,  $E = m \cdot c^2$ , o la fórmula para el cálculo de  $\vec{s} = \vec{v} \cdot t$  embebidas dentro de este párrafo están escritas utilizando este método. Si queremos que la fórmula vaya en una línea aparte la rodearemos de \$\$.

En un texto no científico lo más normal es que utilicemos fórmulas no muy complicadas, en las que básicamente se hace uso de las fracciones, alguna que otra raíz cuadrada, subíndices y potencias. Veamos cómo se escriben en  $\text{\LaTeX}$  mediante algunos ejemplos:

$E = m \cdot c^2$	<code>\$E=m \cdot c^2\$</code>
$\vec{s} = \vec{v} \cdot t$	<code>\$\vec{s}=\vec{v} \cdot t\$</code>
$v_1 = 1 + 3t_1$	<code>\$v_1 = 1 + 3t_1\$</code>
$i = \frac{c \cdot r \cdot t}{100}$	<code>\$i = \frac{c \cdot r \cdot t}{100}\$</code>
$\sqrt{2}$	<code>\$\sqrt{2}\$</code>
$r = \frac{i \cdot 100}{c \cdot t}$	<code>\$r = \frac{i \cdot 100}{c \cdot t}\$</code>

Cuadro 4.1: Fórmulas básicas

El otro método consiste en la utilización del entorno *equation* más apto cuando la fórmula no ha de ir dentro del mismo párrafo del texto y queremos aprovechar las posibilidades de numeración automática de fórmulas de  $\text{\LaTeX}$ .

```
\begin{equation}
ax^2 + bx + c = 0
\rightarrow
x = \frac{-b \pm
\sqrt{b^2 - 4ac}}
{2a}
\end{equation}
```

$$ax^2 + bx + c = 0 \rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (4.1)$$

# Capítulo 5

## Otras fuentes de información

### 5.1. Enlaces

- Kile $\text{\TeX}$   
.....<http://kile.sourceforge.net>
- MiK $\text{\TeX}$   
.....<http://www.miktex.org>
- $\text{\TeX}$ NicsCenter  
.....<http://www.miktex.org>
- CTAN  
.....<http://www.ctan.org>
- latex4wp  
<http://www.ctan.org/tex-archive/help/Catalogue/entries/latex4wp.html>
- Cervan $\text{\TeX}$   
.....<http://www.cervantex.org>
- The not so short introduction to LaTeX2e  
.....<http://people.ee.ethz.ch/~oetiker/lshort/lshort.pdf>

### 5.2. Bibliografía

- **LaTeX: A document preparation system** Libro de Leslie Lamport (autor de LaTeX). Contiene todo lo necesario para iniciarse a este procesador de textos<sup>1</sup>.
- **The LaTeX Companion** Libro de Michel Goossens. Información sobre gran cantidad de los paquetes que usa LaTeX para tareas determinadas: dibujar circuitos electrónicos, caracteres hebreos, dibujar diagramas musicales, autómatas, diapositivas, carátulas de cd...

---

<sup>1</sup>Gracias a la gente de Bulma por algunas referencias



- **LaTeX: una imprenta en sus manos** Libro de Bernardo Cascales Salinas, agrupa ambos libros y en español, una buena alternativa si no te gustan los libros en inglés, es un libro bastante grande y cubre todo lo necesario para utilizar LaTeX desde nivel básico a nivel avanzado.
- **Iniciación a L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>**. Libro de Javier Sanguino Botella. Una buena introducción a L<sup>A</sup>T<sub>E</sub>X destinada fundamentalmente a la composición de textos científicos, aunque en los capítulos finales hace referencia a la utilización de L<sup>A</sup>T<sub>E</sub>X para el tratamiento de otro tipo de textos.