¿Alergia al LaTeX?

Jornadas de presentación de SD:
Procesamiento de textos con LaTeX I

F. Javier Pueyo Mena
kiko
e-mail: kiko@sindominio.net

17 de marzo de 2000

Resumen

Este micro manual está hecho con LaTeX 2ε, un formato especial de TeX pensado para artículos, libros e informes técnicos, en el cual se automatizan numerosas tareas típicas de trabajos científicos: tablas, figuras, ecuaciones, referencias internas, bibliografías, etc. Es especialmente bueno para producir matemáticas, pero ya se está usando más (demasiado poco) en humanidades (curiosamente entregadas, con gran vergüenza, al software y a formatos propietarios).

Índice General

Índice de Tablas

Índice de Figuras

1 Cómo funciona LaTeX 2ε... o la dicha de LaTeX 2εear

1.1 LaTeX tal cual

Supongamos que nunca antes habíamos usado TeX ni LaTeX, y supongamos que nuestra distribución de `Linux` traía TeX, y supongamos que lo habíamos instalado, y supongamos que alguien nos manda un artículo que ha escrito para que lo leamos y le hagamos sugerencias (y supongamos que no es alguien de humanidades y resulta que el artículo no está en Word) y supongamos que se llama `articulo.tex` y supogamos que lo abrimos con nuestro editor y nos encontramos un fichero completamente en texto plano pero con un montón de marcas que empiezan con \ (barra inclinada) y que no entendemos. Dada la extensión del fichero y dadas las marcas juraríamos que está escrito en TeX o en LaTeX... premio.

Podríamos leerlo tal cual, pero sería un poco incómodo, así que nos preguntamos cómo demonios podemos transformar tanta marca en algo legible e imprimible. Digamos que está en LaTeX. Fácil. En nuestra consola tecleamos:

```
NuestraConsola$: latex articulo.tex
```

Y ya está. La primera vez que se usa LaTeX 2ε en `Linux` tardará un poco en procesar el fichero porque estará creando las fuentes estándar de TeX... Pero esto de las fuentes es otra historia y debe ser contada en otra ocasión.

El caso es que el comando `latex` es un atajo. LaTeX no es un programa. El programa que realmente interpreta el texto y las fuentes y tiene los algoritmos para cortar líneas, páginas, palabras y caracteres es TeX, creado por Donald E. Knuth hace mil años con la intención de conseguir que los ordenadores pudieran hacer un trabajo de edición un poco parecido en calidad al de las imprentas manuales. Lo consiguió. LaTeX es lo que se llama un formato, un conjunto de macros —muy complejas— que hacen que TeX sea más fácil de usar y que se automaticen muchas tareas. Así que cuando estamos tecleando `latex` en nuestra consola, en realidad estamos tecleando:

```
NuestraConsola$: tex &latex articulo.tex
```

Es decir, TeXéame `articulo.tex` usando el formato LaTeX. A su vez LaTeX es fácilmente modificable y se puden escribir macros que lo adapten a nuestras necesidades... Pero esto es otra historia y debe ser contada en otra ocasión.

Una vez que hemos LaTeXeado un fichero, TeX normalmente produce otro fichero (en realidad más de uno, pero nos interesa este) con la extensión `dvi`: *device independence*. Es decir, un fichero en un formato estándar que no depende ni de la arquitectura del ordenador que esamos usando ni, por supuesto, del sistema operativo, ni por supuesto de ningún programa en particular, ni de ningún sistema específico de impresión. La filosofía: cualquiera puede usarlo en cualquier sitio con cualquier cosa y obtener el mismo resultado. Cuando a Knuth se le ocurrió lo de estos `dvi` no había inventado ni `ps` ni `pdf` ni nada parecido. Ahora hay versiones especiales de TeX que por defecto producen PostScript o PDF. También hay programas a parte que convierten los `dvi` a estos formatos.

Pero primero lo primero. ¿Cómo veo o imprimo el «output» estándar de LaTeX? Pues depende mucho de los programas que traiga tu distribución de `Linux` y sobre todo de si tienes X Windows o no. Si tienes X y quieres ver directamente el fichero `articulo.dvi` que ha salido de `articulo.tex`, entonces teclea en una XTerm:

```
NuestraXTerm$: xdvi articulo.dvi
```

y lo verás...

Es posible encontrar en la red programas para UNIX que nos permitan ver los `dvi` en la consola, yo los he visto para DOS... Pero eso es otra historia y debe ser contada en otra ocasión...

Pero hay otras posibilidades. La más obvia en sistemas UNIX es producir un fichero PostScript y mandarlo directamente a una impresora PostScript. Pero como la probabilidad de que cualquiera tenga una impresora PS tiende a cero y como la cantidad de árboles que nos están dejando también, es mejor usar un visualizador de PS (que a su vez pueda imprimirlo a cualquier impresora que no sea winprinter), el cual sí pueda tener todo el mundo, porque sea libre: por ejemplo, `ghostscript` y/o `ghostview` que suele venir en todas las distribuciones de `Linux`.

Por pasos. Lo primero que hacemos es crear un fichero `articulo.ps` a partir de nuestro `articulo.dvi` con un programa que viene en cualquier distribución de TeX para `linux`, y que se llama `dvips`:

```
NuestraConsola$: dvips -o articulo.ps articulo.dvi
```

Si asumimos que la opción -o equivale a «output», la línea se explica sola.

Lo segundo es verlo. Usamos `ghostview` y otra vez necesitamos las X:

```
NuestraXTerm$: gv articulo.ps
```

Y otra posibilidad es usar `pdflatex` o `pdftex` en lugar del LaTeX o TeX estándares. Suelen venir en la distribución normal de TeX para `Linux`. Los pasos entonces son:

```
NuestraConsola$: pdflatex articulo.tex
```

Y lo que obtenemos, en lugar de `articulo.dvi` o `articulo.ps` es un lindo gatito: `articulo.pdf`. No está mal, para ser software libre `:)`.

Para verlo o imprimirlo, por ejemplo:

```
NuestraXTerm$: xpdf articulo.pdf
```

Sí, otra vez las X.

1.2 ¡LaTeX enano parla castellano!

A diferencia de los que sólo pueden decir cosas como esta en una sola lengua, TeX y LaTeX (a pesar de ser nativos de inglés) son más que bilingües, son multilingües.

Pero de primeras LaTeX asume que estamos en la parte anglófona de California y hay que decirle que no. Lo más difícil es decirle que use el sistema de división de palabras del castellano o del catalán o del checo o del latín. Y digo que es lo más difícil, porque otras cosas referentes a la lengua las podremos especificar en un documento cualquiera, pero esto de dividir las palabras en sílabas (o en lo que sea) son algoritmos complicados que hay que dárselos a TeX masticados, es decir, compilados. Afortunadamente no hay que recompilar TeX, sino recompilar los formatos que vamos a usar: en nuestro caso LaTeX. Los formatos grandes como LaTeX o `plain` TeX no es obligatorio compilarlos, pero es muy conveniente ya que la velocidad y los recursos consumidos aumentan y disminuyen a la vez. Una vez que una colección de macros está compilada, TeX pone las instrucciones en su memoria sin interpretarlas. Esta compilación se realiza con una versión especial de TeX, que siempre viene con TeX porque es el alma de TeX se llama `IniTeX`, y si `ini` equivale a inicialización, pues el nombre del programa se explica solo.

Por lo general, los formatos como LaTeX o `plain` TeX viene compilados ya, pero vienen sin las reglas de guionado de otras que no sean el inglés, el alemán o el francés. Y nosotros necesitamos otras. Así que necesitamos recompilar nuestro formato de LaTeX y/o nuestro formato de `pdflatex`. Veamos lo que hay que hacer:

Lo primero hay que localizar en nuestra distribución un fichero que se llama, curiosamente, `language.dat`. Si tu distribución de `Linux` incluye te-TeX la estructura de directorios es una estándar de TeX, es decir, la TDS (*TeX Directory Structure*), que significa que el directorio raiz de todo el sistema TeX y `metafont` comenzará a partir del directorio `/texmf`. Los ejemplos que doy siguen la TDS. En fin, editamos `language.dat`, como `root`:

```
NuestraConsola#: emacs /usr/lib/texmf/tex/generic/config/language.dat
```

por ejemplo, y antes de las líneas que dicen:

```
american ushyph1.tex
=USenglish
```

nosotros por el morro, ponemos esto (eso sí en inglés):

```
spanish sphyph.tex
```

o

```
catalan cahyph.tex
```

o los dos. Esto significa dos cosas: primero, que la próxima vez que compilemos LaTeX, se usarán las reglas de división de palabras del castellano o del catalán o de los dos. Los ficheros que contienen estas reglas están en el directorio `/texmf/tex/generic/hyphen`. Y segundo, el hecho de ponerlo(s) en primer lugar, significa que LaTeX usará por defecto esas reglas y no las del americano.

Si hacemos estas modificaciones como `root` en ese directorio, todos los usarios tendrán las reglas de castellano o de catalán por defecto. Si no somos `root` o nuestro sistema es multilingüe, lo mejor es copiar el fichero `language.dat` a nuestra a nuestra `home`, modificarlo allí, salvarlo y dejarlo para seguir los siguientes pasos.

El siguiente paso, es inicializar LaTeX de nuevo. Para ello tenemos que localizar el fichero fuente de LaTeX que se llama `latex.ltx` y luego compilarlo:

```
NuestraConsola$: initex /usr/lib/texmf/tex/latex/base/latex.ltx
```

el resultado será un fichero `latex.fmt`, compilado, en nuestra `home` (bueno en el directorio desde el que

hayamos lanzado `initex`), es decir, el cambio puede ser local a cada `user`. Este fichero será el que lea TEX para procesar nuestros documentos. Si el `root` quiere que este cambio sea para todo el sistema, tendrá que buscar el fichero `latex.fmt` original y pisarlo con el nuevo.

Si lo que vamos a usar es `pdflatex` en lugar del LATEX estándar, entonces lo que tenemos que buscar no es `latex.ltx` sino `pdflatex.ini` y una vez localizado:

```
NuestraConsola$: pdfinitex /usr/lib/texmf/latex/config/pdflatex.ini
```

y obtendremos el fichero `pdflatex.fmt`, y a partir de aquí vale la explicación anterior.

Cómo cambiar de lengua dentro de un mismo documento o cómo modificar las cosas que afectan a la lengua, lo veremos luego, porque... esto es otra historia y debe ser contada en otra ocasión.

2 Anatomía de la bestia: estructura de un documento LATEX

Pero si no te han enviado un fichero y lo que quieres es crearlo tú, entonces estaría bien conocer la estructura normal de un documento LATEX.

Tanto LATEX como TEX utilizan ficheros escritos en ASCII. Antiguamente sólo admitían textos de 7 bits, pero ahora ambos pueden utilizar las 256 posiciones ASCII.

Un documento LATEX tiene dos partes bien definidas: un *preámbulo* y el propio *cuerpo* del texto. El preámbulo contiene indicaciones generales que afectan al documento completo. Por ejemplo, si es un libro o es un artículo, si vas a escribir en varias lenguas y en cuáles, si cambias el formato de la página, etc.

Llamamos preámbulo a todo lo que aparece antes del comando:

```
\begin{document}
```

Hay dos comandos fundamentales en el preámbulo. Estos dos comandos son diferentes en las versiones antiguas de LATEX (la última de las llamadas antiguas fue la LATEX 2.09) y la nueva, es decir, LATEX 2_ε (que es la que usamos nosotros y la que usará todo el mundo en un futuro cercano).

En LATEX 2_ε, los dos comandos con los que comienza cualquier documento son:

```
\documentclass[opciones]{clase}
\usepackage[opciones]{paquete}
```

Es decir, LATEX 2_ε usa *clases* y *paquetes*. La diferencia está en que las *clases* son de rango superior, son obligatorias (de hecho, el comando `\usepackage` es opcional) y sólo puede usarse una en cada documento, ya que afectan al diseño total del documento. Es decir, un documento es durante TODO el documento un *artículo*, un *libro* o un *informe*. Los *paquetes* se usan SOBRE las clases y, por lo general, pueden ser utilizados en cualquiera de ellas. Afectan a cosas más concretas del documento (por ejemplo, un *paquete* puede ser un fichero personal que incluya todas nuestras macros o que incluya nuestras modificaciones personales del tamaño de página, cabecera, etc.)

Por ejemplo, este documento empieza con los siguientes comandos:

```
\documentclass[10pt,a4paper,twoside]{article}
\usepackage{latexsym}
\usepackage[latin1]{inputenc}
\usepackage[spanish]{babel}
```

Es decir, usamos la *clase* `article` con las *opciones* `10pt` (tamaño de letra base de 10 puntos), `a4paper` (tamaño DIN A4, ya que por defecto TEX asume el tamaño de papel americano) y `twoside` (ya que en la clase `article`, por defecto se dan márgenes y cabeceras iguales a las páginas pares e impares: yo paso de los márgenes pero quiero dos cabeceras distintas). Y además, usamos los *paquetes* siguientes: `babel` (paquete estándar de LATEX 2_ε para el uso de varias lenguas) con la *opción* `spanish` (también estándar de LATEX 2_ε para que ciertos títulos automáticos o la fecha aparezcan en español). El paquete `latexsym` para poder usar algunos símbolos matemáticos especiales (los cuales se han sacado del LATEX normal en

la última versión para no cargar el sistema). Y el paquete `inputenc`, que debe significar algo como *input encoding*, es decir, que es un paquete que te permite usar un montón diferente de juegos de asignaciones de caracteres:

- `ascii` ASCII caracteres en el rango 32–127.
- `latin1` ISO Latin-1.
- `latin2` ISO Latin-2.
- `latin3` ISO Latin-3.
- `latin5` ISO Latin-5.
- `decmulti` DEC Multinational Character Set.
- `cp850` IBM 850.
- `cp852` IBM 852.
- `cp437` IBM 437.
- `cp437de` IBM 437 (versión alemana).
- `cp865` IBM 865.
- `applemac` Macintosh.
- `next` Next.
- `ansinew` Windows 3.1 ANSI, extension del Latin-1.
- `cp1250` Windows 1250 (Europa del este y central).

Para el castellano usamos la especificación ISO Latin-1, por lo que nuestra opción será `latin1`.

En la versión antigua LaTeX 2.09, los documentos tenían *styles*, como `article` o `book`, y *options*, como `spanish`, `babel` o `latexsym`.

Es decir, en la versión antigua nuestro documento hubiese empezado:

```
\documentstyle[10pt,a4paper,twoside,latexsym,babel,spanish]{article}
```

LaTeX 2_ε todavía entiende el comando `\documentstyle` de LaTeX 2.09. Pero si lo usamos el programa entra en lo que se llama *LaTeX 2.09 compatibility mode* (modo compatible) y el procesamiento es más lento. Sólo debe usarse si se tienen antiguos documentos LaTeX y no se tiene tiempo (o ganas) de adaptarlos al nuevo estándar.

Una diferencia más está en la extensión de los nuevos ficheros *clase* y los *paquetes*. Las clases ahora tienen extensión `.cls` (antes tenían extensión `.sty`). Los paquetes conservan la extensión `.sty`, ya que prácticamente todos los paquetes de la versión LaTeX 2.09 funcionan a la perfección con LaTeX 2_ε.

Como último ejemplo: si queremos producir un `pdf` y queremos usar las features de color, hipertexto, etc. que permite el formato, añadiremos las opciones siguientes:

```
\documentclass[pdftex]{article}
\RequirePackage[colorlinks,hyperindex]{hyperref}
```

. . . pero esto es otra historia y debe ser contada en otra ocasión.

2.1 *Clases* estándar con mucha clase

Las *clases* siguientes se distribuyen siempre con LaTeX:

article Para escribir artículos. Estructura el documento en secciones, subsecciones, párrafos, etc.

book Para escribir libros. Estructura el documento en partes, capítulos, secciones, etc.

report Para escribir informes técnicos. Parecido a `book` y parecido a `article`.

letter Para escribir cartas o memos.

slides Es el antiguo SLiTeX. Se usa para presentar diapositivas.

proc Para *proceedings*, se basa en la clase `article`.

ltxdoc Para documentar paquetes y clases de LaTeX.

ltxguide El formato usado en las guías de LaTeX 2_ε como, por ejemplo, *usrguide* y *clsguide*, se basa en `article`.

minimal Como su nombre indica, casi no trae nada (sólo ocupa 3 líneas). Simplemente ajusta la altura y la anchura, y define `\normalsize`. Sirve para buscar *bugs* en el código de LaTeX o para usarlo como plantilla si diseñas una *clase* completamente nueva.

2.2 *Paquetes* que no te pesarán

Los *paquetes* siguientes se distribuyen normalmente con LaTeX:

alltt Define el "entorno" (*environment*) `alltt`, que es como `verbatim` excepto que `\`, `{` y `}` tienen su significado normal. Se describe en el fichero `alltt.dtx`.

amslatex El estándar de la American Mathematical Society. Incluye el paquete `amsmath`; contiene todos los comandos necesarios para componer fórmulas de gran complejidad siguiendo el formato de la AMS.

babel Para trabajar con múltiples lenguas.

doc Es el paquete básico para procesar la documentación de los programas en LaTeX. Se describe en `doc.dtx`.

inputenc Permite especificar que *encoding* usa LaTeX.

graphics Incluye el paquete `graphics`, para la inclusión de gráficos producidos en otros programas. También incluye el paquete `color`.

graphpap Define el comando `\graphpaper` usado en el entorno `picture`.

ifthen Permite programar comandos de la forma 'if... then do... otherwise do...' ('si... entonces haz esto... en otro caso haz...'). Se describe en `ifthen.dtx`.

latexsym Como LaTeX 2_ε ya no carga automáticamente la fuente 'symbol' de LaTeX, si se quiere usar hay que llamar al paquete `latexsym`. Se describe en `latexsym.dtx`.

makeidx Define los comandos para producir índices.

showidx Hace que cada comando `\index` se imprima en la página en donde se ha puesto. Está bien para corregir los índices.

tools Varios paquetes escritos por el equipo LaTeX3.

2.3 *Preámbulos* ambulantes

Hay cosas típicas que suelen aparecer en los preámbulos de LaTeX. Un experto LaTeXeador reducirá al mínimo los comandos del preámbulo. Mi consejo es que sólo se incluyan ahí las cosas más particulares del documento en cuestión. Las macros o formatos que se usen habitualmente para más de un documento deben agruparse en ficheros de estilos (`*.sty`) y ser llamados con el comando `\usepackage` (o también `\input`, si el fichero no tiene extensión `.sty`).

Si nos fijamos en el preámbulo de este documento, veremos que hay muchas cosas que no deberían estar ahí. La figura

Figura 1: Preámbulo de este artículo

```
% tamaño de página
   \setlength{\textwidth}{159.2mm}
   \setlength{\oddsidemargin}{-0.04cm}
   \setlength{\topmargin}{-0.54cm}
   \setlength{\textheight}{242mm}
% sangría y separación entre párrafos
    \setlength{\parindent}{0pt}
    \setlength{\parskip}{5pt plus 2pt minus 1pt}
% cabeceras
    \makeatletter % comando que permite usar la @ arroba en las definiciones

    \renewcommand{\@oddhead}{\textit{>Alergia al \LaTeX?}\hfil\textrm{\thepage}}%
    \renewcommand{\@evenhead}{\textrm{\thepage}\hfil\textit{F.~Javier Pueyo
                                                        Mena - kiko}}%
    \renewcommand{\@oddfoot}{}                % vacío
    \renewcommand{\@evenfoot}{\@oddfoot}      % vacío

% Macro del tipo {decl} para los ejemplos de este articulo

    \newenvironment{ejemplos}[1][]%
        {\par\small\addvspace{4.5ex plus 1ex}%
         \vskip -\parskip
         \ifx\relax#1\relax
            \def\@decl@date{}%
         \else
            \def\@decl@date{\NEWfeature{#1}}%
         \fi
         \noindent\hspace{-\leftmargini}%
         \begin{tabular}{|l|}\hline\ignorespaces}%
        {\\\hline\end{tabular}\nobreak\@decl@date\par\nobreak
         \vspace{2.3ex}\vskip -\parskip}

    \makeatother % ya no necesito la @

% Macro para un texto con borde
    \newsavebox{\fmbox}
       \newenvironment{fmpage}[1]
         {\begin{lrbox}{\fmbox}\begin{minipage}{#1}}
         {\end{minipage}\end{lrbox}\fbox{\usebox{\fmbox}}}

% macro para mostrar los ejemplos enfrentados
    \newlength{\egwidth}\setlength{\egwidth}{0.50\textwidth}
    \newenvironment{eg}%
      {\begin{list}{}{\setlength{\leftmargin}{0\textwidth}%
      \setlength{\rightmargin}{\leftmargin}}\item[]\footnotesize}%
      {\end{list}}
    \newenvironment{egbox}%
      {\begin{minipage}[t]{\egwidth}}%
      {\end{minipage}}
    \newcommand{\egstart}{\begin{eg}\begin{egbox}}
    \newcommand{\egmid}{\end{egbox}\hfill\begin{egbox}}
    \newcommand{\egend}{\end{egbox}\end{eg}}
% Asignaciones para que los floats floten menos
    \setcounter{topnumber}{5}
    \setcounter{bottomnumber}{5}
    \setcounter{totalnumber}{10}
    \renewcommand{\topfraction}{.9}
    \renewcommand{\bottomfraction}{.9}
    \renewcommand{\textfraction}{.05}
    \renewcommand{\floatpagefraction}{.05}
```

2.4 Sin más *preámbulos*

Una vez que hemos especificado todas nuestras preferencias, paquetes, macros, etc. en el preámbulo, insertamos el comando:

```
\begin{document}
```

que—como su propio nombre indica—supone el verdadero comienzo de nuestro texto.

Una vez hayamos escrito todo lo que queramos, terminaremos el documento con el comando:

```
\end{document}
```

Todo lo que se escriba debajo de dicho comando será ignorado por TEX.

Pero antes de comenzar a escribir el texto en sí, podemos especificar unas cuantas cosas: el título del artículo o libro, el autor, la fecha, si debe haber un índice de contenidos, de tablas, de figuras, etc.

Por ejemplo, en este artículo—justo debajo de `\begin{document}`—hemos escrito lo que aparece en la Figura

Figura 2: Encabezamiento de este artículo

```
\title{Alergia al \LaTeX\\[.5cm]                                          % Título
      {\itshape {\Large  Jornadas de presentación de SD:\\               % subtitulo
      Procesamiento de textos con \LaTeX\ I}}}

\author{F.~Javier Pueyo Mena\\                                  % Autor
        \textit{kiko}\\
        \textit{e-mail}: \texttt{kiko@sindominio.net}}          % mail

\date{\today}       % Fecha de (hoy) escritura del artículo,
                    % saldrá en castellano gracias a BABEL.

\maketitle          % Este comando imprime lo anterior

\begin{abstract}
Este micro manual está hecho con \TeX\ un formato ...  y a formatos propietarios).
\end{abstract}

\tableofcontents    % Este comando imprime el índice de contenidos
\listoftables       % Este comando imprime el índice de tablas
\listoffigures      % Este comando imprime el índice de figuras
```

Es decir, hemos utilizado los comandos:

```
\title{ }
\author{ }
\date{ }
```

que recogen la información. Y después hemos usado el comando:

```
\maketitle
```

que hace que esa información se imprima con un formato predefinido. Después hemos destacado un 'abstract' o resumen con el entorno:

```
\begin{abstract}
\end{abstract}
```

Finalmente introducimos un comando que indica a LaTeX 2_ε que vamos a querer un índice de contenidos, que incluya automáticamente nuestros capítulos, secciones, etc. con su título y número de página. Los

otros dos comandos imprimen los índices de tablas y de figuras. El uso de estos comandos obliga a procesar nuestro documento un mínimo de dos veces, para asegurar que en el índice aparecen correctamente los números de sección y las páginas. Cuando esta segunda (o tercera) vuelta es necesaria, LaTeX 2$_\varepsilon$ informa de ello. Estos comandos pueden situarse en cualquier parte del documento que queramos que aparezcan los índices.

```
\tableofcontents
\listoftables
\listoffigures
```

Después de estos comandos—ahora sí—podemos empezar a escribir nuestro trabajo.

2.5 Del todo a las partes

Un documento-libro (book) o un documento-informe (report) de LaTeX se estructura en partes, capítulos, secciones, subsecciones y subsubsecciones. Si es un documento-artículo (article) no necesitamos (ni tenemos disponibles) las partes ni los capítulos. Los siguientes comandos se usan para indicar las divisiones:

```
\part{Título de la parte}
\chapter{Título del capítulo}
\section{Título de la sección}
\subsection{Título de la subsección}
\subsubsection{Título de la subsubsección}
```

Las consecuencias de usar estos comandos son varias:

1ª Disparar e ir actualizando un contador para cada una de las divisiones.

2ª Imprimir el número de capítulo (sección, subsección, etc.) seguido del título. Dependiendo de la jerarquía de las divisiones irá aumentando o disminuyendo el tamaño de la letra, y cambiará a **negrita**. Estos cambios en el formato de los títulos están especificados en la *clase* (por ejemplo en `article.cls`). Se pueden cambiar fácilmente.

3ª Incluir la sección en el índice de contenidos.

Si queremos que una de las divisiones no sea numerada (y por lo tanto no entre en el índice de contenidos) tendremos que usar las siguientes variantes (variantes con *) de los comandos:

```
\part*{Título de la parte}
\chapter*{Título del capítulo}
\section*{Título de la sección}
\subsection*{Título de la subsección}
\subsubsection*{Título de la subsubsección}
```

Una división más que se puede establecer es la de párrafo (y subpárrafo):

```
\paragraph{Título del párrafo}
\subparagraph{Título del subpárrafo}
```

Este es el título de este párrafo: Este comando simplemente sirve para destacar un título de párrafo pero no imprime contadores ni aparece en el índice de contenidos. Como su nombre indica su ámbito se reduce al párrafo siguiente al comando. Por supuesto, no hay que usar este comando para los párrafos normales en LaTeX. Los párrafos normales se marcan dejando una línea en blanco entre uno y otro.

Este es el título de este subpárrafo: Como vemos, el comando `\subparagraph{ }` actúa exactamente igual que el anterior comando de `\paragraph{ }`.

La diferencia entre ambos es jerárquica. Si colocamos en el preámbulo del documento el comando:

```
\setcounter{tocdepth}{5}
```

conseguiremos que los párrafos y subpárrafos queden incluidos en el índice de contenidos, y se sangren jerárquicamente.

Un punto a favor de usar las divisiones de LaTeX 2ε es que después podemos referirnos a ellas fácilmente sin necesidad de saber el número que TeX les has asignado. Esto se hace mediante los comandos:

```
\label{ }
\ref{ }
```

El comando `\label{ }` se coloca justo detrás del título de sección, por ejemplo:

```
\section{Biblias medievales romanceadas}\label{biblias}
```

y después, cuando queramos referirnos a esta sección en el texto (ya sea antes o después de dicha sección) escribiremos, por ejemplo:

```
En la seccion~\ref{biblias} trataremos de las Biblias medievales
```

y obtendremos algo parecido a:

En la sección 4.1 trataremos de las Biblias medievales...

Llamo la atención sobre el uso de `~` delante del comando `\ref`. Es conveniente ponerlo (aunque no obligatorio) para que el número de sección no se separe nunca de la palabra "sección".

Los dos comandos explicados pueden usarse no sólo en la referencia a las divisiones de LaTeX 2ε sino también para referir a tablas y figuras. Esto lo veremos en las secciones

La última frase de este párrafo decía en mi original:

```
Esto lo veremos en las secciones~\ref{table} y~\ref{fig}.
```

3 Y se hizo ... la letra

Escribir un fichero para TeX o para LaTeX requiere acostumbrarse a ciertas convenciones y usos que no son necesarios en los procesadores de texto WYSIWYG[1].

3.1 Caracteres muy especiales

Hay una serie de caracteres que en principio—pero sólo en principio—están reservados para ciertas funciones. Los fondos (perdón...) reservados son:

```
~  #  $  %  ^  &  {  }  \  _
```

Para poder escribirlos tenemos que usar los siguientes comandos:

```
\~{}  \#  \$  \%  \^{}  \&  \{  \}  $\backslash$  \_
```

y obtendremos:

~ # $ % ^ & { } \ _

—¿Para qué sirven esos caracteres entonces?
—Me alegro de que me haga esa pregunta:

`~`: es un pequeño comando que colocado entre dos palabras, en lugar de un espacio, impide que estas dos palabras se separen en distintas líneas. Se suele usar en casos como `Sr.~Lazaro` (Sr. Lázaro) en los que queda francamente mal separar a tan ilustre personajillo de su títulillo y distinción.

[1] WYSIWYG: *What you see is what you get* o también WYSIAYG: *What you see is ALL you get*

`#`: se usa como parámetro en las macros.

`$`: se usa para entrar en (y salir de) *modo matemático*, es decir, para escribir fórmulas, ecuaciones, etc.

`%`: en un documento de TeX se usa para hacer comentarios sin que se impriman. Todo lo que está a la derecha de `%` (en la misma línea) no será procesado por TeX.

`^`: se usa como superíndice (123^{456}).

`&`: se usa como signo de tabulación en las tablas.

`{ }`: las dos llaves se usan para limitar el alcance de los comandos de TeX. O para hacer grupos, que no sé si es lo mismo, pero me da igual.

`\`: es, por defecto, el caracter de escape en TeX y en LaTeX, es decir, el que indica que lo que sigue es un comando.

`_`: se usa como subíndice (123_{456}).

3.2 La cartilla en LaTeX 2ε

En TeX Las letras sin acentuar se escriben tal cual (menos mal):

```
a b c d e f g h i j k l m o p q r s u w x y z
```

y tenemos:

a b c d e f g h i j k l m o p q r s u w x y z

Los espacios y los tabuladores son ignorados por TeX normalmente, es decir, que da lo mismo que escribas un espacio que 100, aunque el original sufriría bastante. También da igual un retorno aquí y allá, los párrafos se marcan con una línea (o más) en blanco:

```
Estos      espacios
    y este retorno no se ven en el output

Pero la línea en blanco anterior crea este nuevo párrafo
```

que produce:

Estos espacios y este retorno no se ven en el output.

Pero la línea en blanco anterior crea este nuevo párrafo.

Como decíamos TeX habla inglés por defecto, y esto se aplica también a las reglas tipográficas. Una de estas reglas afecta a los espacios extras que los anglosajones dejan detrás de los signos de puntuación. En las demás tradiciones tipográficas esto no es así, por lo que es muy conveniente cambiar este comportamiento por defecto de TeX. La mejor manera de hacerlo es desabilitando de un golpe esta función. El comando que hay que colocar al pricipio del documento es:

```
\frenchspacing
```

y curiosamente para volver al comportamiento normal de TeX hay que escribir:

```
\nonfrenchspacing
```

Otro comportamiento peculiar de TeX es que para conseguir la comillas dobles, no hay que escribir las comillas dobles del teclado (¿quién dijo que TeX era fácil?), sino escribir dos comillas simples de apertura (es decir, en el teclado español, dos acentos graves) seguidas para las dobles de apertura, y dos comillas simples de cierre para las dobles de cierre. Las simples son más simples, son lo que son:

```
Estas ‘‘comillas dobles’’ producen estas:
Estas “comillas dobles” producen estas.
Estas ‘comillas simples’ producen estas:
Estas ‘comillas simples’ producen estas.
```

Respecto a los caracteres acentuados y signos especiales tenemos un pequeño problema: si no usamos la versión multilingüe de `plain` TEX, o no usamos el paquete `inputenc` de LATEX, entonces tenemos que usar comandos para producir caracteres con diacríticos. Lo cual para alguien que escribe eñes y acentos por un tubo es molesto. Pero si usamos esos paquetes podremos escribir los caracteres tal cual en nuestro fichero. Ahora, conocer los comandos estándar de LATEX está muy bien si no tenemos acceso a un teclado español o si queremos usar algo tan simple como una jota sin punto: ȷque evidentemente no está en el teclado. Los tipos de acento que hay en TEX son los siguientes (se puede subsistuir la [o] por cualquier letra, excepto la [i] y la [j] a las que primero hay que quitarles el punto de esta manera:

```
\i \j
```

y tenemos:

```
ı ȷ
```

Con los comandos:

```
\’o
\’{\i}
\’{\j}
\‘o
\^o
\" o
\~o
\.o
\v o
\u o
\H o
\c o
\b o
\d o
\t oo
```

obtendremos:

```
ó í ȷ́ ò ô ö õ ȯ ǒ ǒ ő o̧ o̲ ọ o͡o
```

Estamos asumiendo que estamos usando las fuentes estándar de TEX, es decir, las CMR o *Computer Modern Roman*, inventadas, diseñadas y dibujadas por el creador de TEX, y que siempre vienen con TEX. Estas fuentes tienen su propia asignación de caracteres e incluyen los diacríticos que he mostrado arriba. TEX no está limitado a usar estas fuentes, puede usar `ttf` o `ps` ... pero esa es otra historia y debe ser contada en otra ocasión. El caso es que las CMR están muy bien y con ellas puedes hacer cosas como esta:

```
w̌ x̂y š ḥ
```

Otros símbolos especiales se logran con los comandos:

```
\oe
\OE
\ae
\AE
\aa
\AA
\o
\O
\l
\L
\ss
\dag
\ddag
\S
\P
\copyright
\pounds
```

que producen:

œ Œ æ Æ å Å ø Ø ł Ł ß † ‡ § ¶ © £

En modo matemático—al que se entra tecleando $ y se cierra con otro $—se pueden obtener muchos más símbolos especiales (estos son válidos para cualquier fuente ya que en realidad pertenecen a una fuente específica de TeX que no es CMR):

He aquí los comandos que producen las letras griegas en minúscula:

```
$\alpha$  $\beta$  $\gamma$  $\delta$  $\epsilon$  $\varepsilon$  $\zeta$  $\eta$
$\theta$  $\vartheta$  $\iota$  $\kappa$  $\lambda$  $\mu$  $\nu$  $\xi$
$\pi$  $\varpi$  $\rho$  $\varrho$  $\sigma$  $\varsigma$  $\tau$  $\upsilon$
$\phi$  $\varphi$  $\chi$  $\psi$  $\omega$
```

Comandos que producen este resultado:

α β γ δ ϵ ε ζ η θ ϑ ι κ λ μ ν ξ π ϖ ρ ϱ σ ς τ υ ϕ φ χ ψ ω

Y estas son algunas mayúsculas:

```
$\Gamma$  $\Delta$  $\Theta$  $\Lambda$  $\Xi$  $\Pi$
$\Sigma$  $\Upsilon$  $\Phi$  $\Psi$  $\Omega$
```

que prodecen lo siguiente:

Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω

Para conseguir los tres tipos de guiones:

```
-
--
---
```

combinaciones de guión simple que producen:

-
–
—

Para lograr la interrogación y la exclamación de apertura se usan los comandos:

```
?`
!`
```

es decir, las de cierre más una comilla simple de apertura (acento grave en el teclado español) que producen:

```
¿
¡
```

Pero queda dicho: si quieres puedes poner directamente una interrogación (¿) o una exclamación (¡) o una comilla angular (≪) o una a acentuada (á) o lo que quieras. Si te encontraras con un mensaje de error en algunos de esos símbolos, significa que ese símbolo no está declarado en el fichero `latin1.def` y tienes que definirlo tú. Por ejemplo yo uso en este artículo tres símbolos que no están definidos: los guiones medio (–) y largos (—) y la cruz llamada ≪dag≫. Lo que he hecho es añadir en el preámbulo las tres líneas siguientes:

```
\DeclareInputText{151}{---}
\DeclareInputText{150}{--}
\DeclareInputText{134}{\dag}
```

el comando `\DeclareInputText{}{}` toma dos argumentos: el primero la posición ISO o ASCII o lo que sea del caracter que queremos teclear tal cual en nuestros ficheros y el segundo lo que queremos que haga TEX cuando se encuentre este caracter.

3.3 Cursivas, negritas y otras especies

En esta sección veremos cómo—dentro de una misma fuente—se puede seleccionar *cursiva*, **negrita**, `fuente fija` o sans serif.

En las palabras anteriores hemos usado el antiguo sistema de comandos de LATEX, pero se recomienda usar el nuevo estándar de LATEX 2_ε.

El sistema antiguo utiliza los comandos:

```
{\it texto en cursiva}
{\bf texto en negrita}
{\tt texto en fuente fija}
{\sf texto en tipo sans-serif}
```

para producir los efectos:

texto en cursiva
texto en negrita
`texto en fuente fija`
texto en tipo sans-serif

Pero estos comandos no funcionaban muy intuitivamente por lo que el equipo de LATEX3 los ha rehecho y les ha cambiado los nombres para no liarnos. Un problema típico con los comandos antiguos era este ejemplo:

```
{\it texto en cursiva y esto debería ser {\bf cursiva y negrita} a la vez}
```

texto en cursiva y esto debería ser intuitivamente **cursiva y negrita** *a la vez... pero evidentemente sólo es negrita.*

Los nuevos comandos desde LATEX 2_ε son los siguientes:

```
\textrm{...} o {\rmfamily ...} redonda
\textit{...} o {\itshape ...} cursiva
\textbf{...} o {\bfseries ...} negrita
\texttt{...} o {\ttfamily ...} fuente fija
\textsf{...} o {\sffamily ...} sans serif (tipo arial o helvética)
```

Que usados con el ejemplo anterior, cambian el resultado:

```
{\itshape texto en cursiva y esto debería ser \textbf{cursiva y negrita} a la vez}
```

*texto en cursiva y esto debería ser intuitivamente **cursiva y negrita** a la vez... y lo es.*

Algunos comandos más:

```
\textup{...} o {\upshape ...} upright (tipo dunhill)
\textsf{...} o {\sffamily ...} sans serif (tipo arial)
\textsl{...} o {\slshape ...} slanted (redonda inclinada)
\textsc{...} o {\scshape ...} versalita

\emph{...}   o {\em ...} redonda,  cursiva o subrayado dependiendo del contexto.
```

Un ejemplo de todas las posibilidades:

Redonda: `\textrm{...}`
Esto es texto normal
`{\rmfamily ...}`
Esto también es texto normal

Sans-serif: `\textsf{...}`
Esto es texto en Sans Serif
`{\sffamily ...}`
Esto también es texto en Sans Serif

Fuente fija: `\texttt{...}`
`Esto es texto de fuente fija`
`{\ttfamily ...}`
`Esto también es texto de fuente fija`

Negrita: `\textbf{...}`
Esto es texto en negrita
`{\bfseries ...}`
Esto también es texto en negrita

Cursiva: `\textit{...}`
Esto es texto en cursiva
`{\itshape ...}`
Esto también es texto en cursiva

Énfasis: `\emph{...}`
Esto es texto enfatizado. En este caso cursiva
`{\em ...}`
Esto también es texto enfatizado. En este caso cursiva

Las dos siguientes sólo tienen sentido cuando usamos fuentes que tienen estos diseños, como las CMR:

Slanted : (redonda inclinada)
`\textsl{...}`
Esto es texto inclinado, pero no cursiva
`{\slshape ...}`
Esto también es texto inclinado, pero no cursiva

Versalitas: `\textsc{...}`
Esto es texto en versalitas
`{\scshape ...}`
Esto también es texto en versalitas

3.4 El tamaño de las pequeñas bestias

Respecto a los tamaños: en LaTeX 2$_\varepsilon$ se controlan con los siguientes comandos, de menor a mayor:

```
\tiny
\scriptsize
\footnotesize
\small
\normalsize
\large
\Large
\LARGE
\huge
\Huge
```

Dependiendo de la opción de tamaño del `\documentclass` que se esté usando estos tamaños son relativos al tamaño de la fuente base de dicho `\documentclass`.

Por ejemplo, si se dice:

```
\documentclass[12pt]{book}
```

(en la versión antigua-pero compatible-de LaTeX: `\documentstyle[12pt]{book}`)

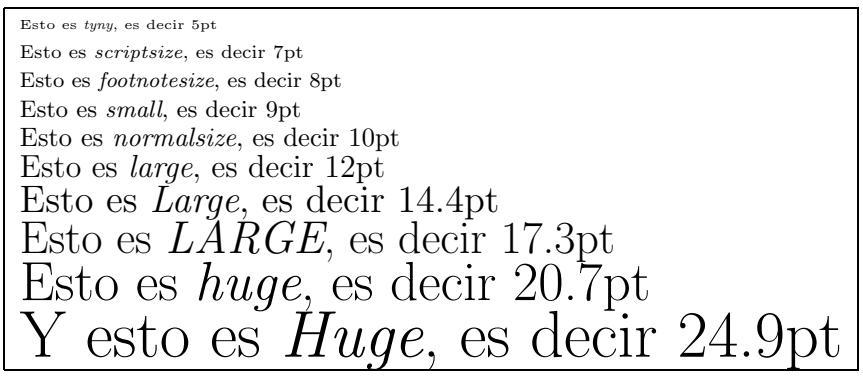
el valor de `\Large` es 17.3pt

pero si se dice:

```
\documentclass[11pt]{book}
```

el valor de `\Large` es 14.4pt

Para nuestro documento, que está en 10pt, los comandos de tamaño preestablecidos dan los siguientes valores:



Aunque en LaTeX 2$_\varepsilon$ también se puede especificar directamente el tamaño mediante el comando:

```
\fontsize{12}{12}\selectfont
```

en donde el primer número es el tamaño de la fuente en "puntos" (12pt) y el segundo (que es obligatorio también) es la interlínea medida en "puntos" (12pt).

Ahí van algunos ejemplos:

```
\fontsize{5}{6}\selectfont
```

Esto es texto en 5 puntos

```
\fontsize{12}{12}\selectfont
```

Esto es texto en 12 puntos

```
\fontsize{16}{14}\selectfont
```

Esto es texto en 16 puntos

```
\fontsize{20}{16}\selectfont
```

Esto es texto en 20 puntos

```
\fontsize{28}{20}\selectfont
```

Esto es texto en 28 puntos

Para volver al tamaño base del documento escribiremos el comando:

```
\normalsize
```

Por defecto LaTeX usa una interlínea de 1 en sus clases estándar. Para cambiar esta interlínea para todo el documento (o para una parte) tenemos que usar el comando `\linespread{`*factor*`}`. Por ejemplo, el comando en el preámbulo:

```
\linespread{1.3}
```

cambiará la interlínea a uno y medio, mientras que:

```
\linespread{1.6}
```

lo cambiará a dos.

4 Por un mejor *entorno*

Una vez que ya sabemos el alfabeto LaTeX 2_εtiano, bueno será meternos un poco en su sintaxis. Una parte importante de ésta son los llamados *entornos* (enviroment).

En LaTeX 2_ε los entornos son pequeñas islas (a veces parecen continentes) en los que se cambian formatos, tipos de letra, etc. temporalmente para conseguir automáticamente efectos distintos.

En esto conoceréis que soy un entorno:

```
\begin{nombre del entorno}
\end{nombre del entorno}
```

por ejemplo:

```
\begin{table}
\end{table}
```

Todo lo que hay entre estos dos comandos obedece a nuevas reglas, fijadas en macros (en este caso hechas por el equipo de LaTeX 2_ε pero podemos hacer nuestros propios entornos) que se anulan después del comando `\end{ }`.

En las secciones siguientes veremos los entornos más comunes, pero hay muchos más. Las hemos ordenado alfabéticamente, pero se pueden reunir según su función de la siguiente forma: para la justificación parcial de fragmentos de texto véanse las subsecciones

4.1 center

Sirve para **centrar** texto. Si no utilizamos el indicador de fin de línea \\, LaTeX 2_ε la romperá donde considere necesario. En este ejemplo usamos \\, en los de las subsecciones

```
Esto es texto sin centrar.
El fragmento corresponde
a los tres primeros versiculos
del Cantar de los Cantares
en la Biblia 10 288.

\begin{center}
(1) Cantar de los cantares de Salamon.\\
(2) [B]esseme de los bessos de su boca,\\
ca mejores son tus amores que el vino.\\
(3) Al olor de tus unguentos buenos,\\
unguento vaziadizo es el tu nonbre,\\
por esso las donzellas te amaron.
\end{center}

Esto es texto sin centrar.
El fragmento corresponde
a los tres primeros versiculos
del Cantar de los Cantares
en la Biblia 10 288.
```

Esto es texto sin centrar. El fragmento corresponde a los tres primeros versículos del Cantar de los Cantares en la Biblia 10 288.

(1) Cantar de los cantares de Salamón.
(2) [B]ésseme de los bessos de su boca,
ca mejores son tus amores que el vino.
(3) Al olor de tus ungüentos buenos,
ungüento vaziadizo es el tu nonbre,
por esso las donzellas te amaron.

Esto es texto sin centrar. El fragmento corresponde a los tres primeros versículos del Cantar de los Cantares en la Biblia 10 288.

4.2 description

Como su propio nombre no indica, se usa para hacer una lista sin numerar, destacando en negrita y con sangría francesa el ítem descrito. Tanto éste como los entornos descritos en las subsecciones

```
\item[ ]
```

que se antepone a cada miembro de la lista. El argumento opcional `[ ]`, se usa para colocar un símbolo delante de un ítem determinado. Si queda vacío, cada entorno de lista tiene símbolos definidos por defecto.

Veamos un ejemplo del entorno `description`:

```
\begin{description}

\item[Primer caso: ] Este es el primer
caso. No se me ocurre nada que decir,
pero como quiero mostrar un texto
largo\dots

\item[Segundo caso: ] Este es el segundo
caso. Sigue sin salirme nada que decir,
pero insisto en mostrar un texto
largo\dots

\item[Tercer caso: ] Este es el tercer
caso. Pues no hay manera, no se me
ocurre nada, pero me obceco en mostrar
un texto largo\dots

\end{description}
```

Primer caso: Este es el primer caso, No se me ocurre nada que decir, pero como quiero mostrar un texto largo...

Segundo caso: Este es el segundo caso. Sigue sin salirme nada que decir, pero insisto en mostrar un texto largo...

Tercer caso: Este es el tercer caso. Pues no hay manera, no se me ocurre nada, pero me obceco en mostrar un texto largo...

4.3 enumerate

Otro de los entornos para hacer listas, en este caso numeradas. Para que se vea como funciona haremos tres niveles, metiendo un entorno en otro. Se observará que por defecto LaTeX numera el primer nivel con números arábigos, el segundo con letras minúsculas, y el tercero con números romanos, ¡qué bien!.

```
\begin{enumerate}

\item Este es el primer caso,
No se me ocurre nada que decir,
pero como quiero mostrar un texto
largo\dots
 \begin{enumerate}
 \item Este el primer item
 del segundo nivel
   \begin{enumerate}
   \item Este el primer item
    del tercer nivel
    \item Este el segundo item
    del tercer nivel
    \item Este el tercer item
    del tercer nivel
    \end{enumerate}
 \item Este el segundo item
 del segundo nivel
   \begin{enumerate}
   \item Este el primer item
    del tercer nivel
    \item Este el segundo item
    del tercer nivel
    \item Este el tercer item
    del tercer nivel
    \end{enumerate}
 \end{enumerate}

\item Este es el segundo caso.
Sigue sin salirme nada que decir,
pero insisto en mostrar un texto
largo\dots

\item Este es el tercer caso.
Pues no hay manera, no se me ocurre nada,
pero me obceco en mostrar un texto
largo\dots

\end{enumerate}
```

1. Este es el primer caso, No se me ocurre nada que decir, pero como quiero mostrar un texto largo...
   (a) Este el primer item del segundo nivel
      i. Este el primer item del tercer nivel
      ii. Este el segundo item del tercer nivel
      iii. Este el tercer item del tercer nivel
   (b) Este el segundo item del segundo nivel
      i. Este el primer item del tercer nivel
      ii. Este el segundo item del tercer nivel
      iii. Este el tercer item del tercer nivel
2. Este es el segundo caso. Sigue sin salirme nada que decir, pero insisto en mostrar un texto largo...
3. Este es el tercer caso. Pues no hay manera, no se me ocurre nada, pero me obceco en mostrar un texto largo...

4.4 figure

No hace nada especial. Simplemente permite mantener un contador de figuras, cuyos títulos pueden ir a un índice de figuras. También permite referirse a una figura con los comandos `\label` y `\ref`. He aquí el ejemplo:

```
\begin{figure}[h]
\begin{center}
\begin{fmpage}{2.5cm}
NO HAY FOTO
\end{fmpage}
\end{center}
\caption{NO HAY FOTO? Este titulillo va
debajo de la figura}\label{titulo}
\end{figure}
```

NO HAY FOTO

Figura 3: ¿NO HAY FOTO? Este titulillo va debajo de la figura

Con lo que ya podemos referirnos a la figura

4.5 flushleft

Sirve para justificar a la **izquierda** el texto. Se usa de la siguiente manera:

```
Esto es texto sin justificar
a la izquierda.
El fragmento corresponde
a los tres primeros versiculos
del Cantar de los Cantares
en la Biblia 10 288.

\begin{flushleft}
(1) Cantar de los cantares de Salamon.
(2) [B]esseme de los bessos de su boca,
ca mejores son tus amores que el vino.
(3) Al olor de tus unguentos buenos,
unguento vaziadizo es el tu nonbre,
por esso las donzellas te amaron.
\end{flushleft}

Esto es texto sin justificar
a la izquierda.
El fragmento corresponde
a los tres primeros versiculos
del Cantar de los Cantares
en la Biblia 10 288.
```

Esto es texto sin justificar a la izquierda. El fragmento corresponde a los tres primeros versículos del Cantar de los Cantares en la Biblia 10 288.

(1) Cantar de los cantares de Salamón. (2) [B]ésseme de los bessos de su boca, ca mejores son tus amores que el vino. (3) Al olor de tus ungüentos buenos, ungüento vaziadizo es el tu nonbre, por esso las donzellas te amaron.

Esto es texto sin justificar a la izquierda. El fragmento corresponde a los tres primeros versículos del Cantar de los Cantares en la Biblia 10 288.

4.6 flushright

Sirve para justificar el texto a la **derecha**. Se usa de la siguiente manera:

```
Esto es texto sin justificar
a la derecha.
El fragmento corresponde
a los tres primeros versiculos
del Cantar de los Cantares
en la Biblia 10 288.

\begin{flushright}
(1) Cantar de los cantares de Salamon.
(2) [B]esseme de los bessos de su boca,
ca mejores son tus amores que el vino.
(3) Al olor de tus unguentos buenos,
unguento vaziadizo es el tu nonbre,
por esso las donzellas te amaron.
\end{flushright}

Esto es texto sin justificar
a la derecha.
El fragmento corresponde
a los tres primeros versiculos
del Cantar de los Cantares
en la Biblia 10 288.
```

Esto es texto sin justificar a la derecha. El fragmento corresponde a los tres primeros versículos del Cantar de los Cantares en la Biblia 10 288.

(1) Cantar de los cantares de Salamón. (2) [B]ésseme de los bessos de su boca, ca mejores son tus amores que el vino. (3) Al olor de tus ungüentos buenos, ungüento vaziadizo es el tu nonbre, por esso las donzellas te amaron.

Esto es texto sin justificar a la derecha. El fragmento corresponde a los tres primeros versículos del Cantar de los Cantares en la Biblia 10 288.

4.7 itemize

He aquí otro de los entornos para crear listas. Nótese en el ejemplo que hemos utilizado la opción `[ ]` en los ítemes 2 y 3 para cambiar el símbolo por defecto, que no está de más decirlo es `$\bullet$`, es decir, •.

```
\begin{itemize}

\item Este es el primer caso,
No se me ocurre nada que decir,
pero como quiero mostrar un texto
largo\dots

\item[$\Box$] Este es el segundo
caso. Sigue sin salirme nada que
decir, pero insisto en mostrar un
texto largo\dots

\item[$\rightarrow$] Este es el tercer
caso. Pues no hay manera, no se me
ocurre nada,pero me obceco en mostrar
un texto largo\dots

\end{itemize}
```

- Este es el primer caso, No se me ocurre nada que decir, pero como quiero mostrar un texto largo...
- □ Este es el segundo caso. Sigue sin salirme nada que decir, pero insisto en mostrar un texto largo...
- → Este es el tercer caso. Pues no hay manera, no se me ocurre nada,pero me obceco en mostrar un texto largo...

Pero no nos conformamos con un nivel sino que queremos más, por lo menos 3. Nótese que en el "segundo caso" también hemos cambiado los símbolos por defecto (- y *) en los niveles más profundos, por ¶ y †:

- Este es el primer caso, No se me ocurre nada que decir, pero como quiero mostrar un texto largo...
  - Este el primer item del segundo nivel
    - Este el primer item del tercer nivel
    - Este el segundo item del tercer nivel
    - Este el tercer item del tercer nivel
  - Este el segundo item del segundo nivel
    - Este el primer item del tercer nivel
    - Este el segundo item del tercer nivel
    - Este el tercer item del tercer nivel

□ Este es el segundo caso. Sigue sin salirme nada que decir, pero insisto en mostrar un texto largo...

¶ Este el primer item del segundo nivel

† Este el primer item del tercer nivel

† Este el segundo item del tercer nivel

† Este el tercer item del tercer nivel

¶ Este el segundo item del segundo nivel

† Este el primer item del tercer nivel

† Este el segundo item del tercer nivel

† Este el tercer item del tercer nivel

```
\begin{itemize}

\item Este es el primer caso,
No se me ocurre nada que decir,
pero como quiero mostrar un texto
largo\dots
 \begin{itemize}
 \item Este el primer item
 del segundo nivel
   \begin{itemize}
   \item Este el primer item
    del tercer nivel
    \item Este el segundo item
    del tercer nivel
    \item Este el tercer item
    del tercer nivel
    \end{itemize}
 \item Este el segundo item
 del segundo nivel
   \begin{itemize}
   \item Este el primer item
    del tercer nivel
    \item Este el segundo item
    del tercer nivel
    \item Este el tercer item
    del tercer nivel
    \end{itemize}
 \end{itemize}

\item[$\Box$] Este es el segundo
caso. Sigue sin salirme nada que
decir, pero insisto en mostrar un
texto largo\dots
 \begin{itemize}
 \item[\P] Este el primer item
 del segundo nivel
   \begin{itemize}
   \item[\dag] Este el primer item
    del tercer nivel
    \item[\dag] Este el segundo item
    del tercer nivel
    \item[\dag] Este el tercer item
    del tercer nivel
    \end{itemize}
 \item[\P] Este el segundo item
 del segundo nivel
   \begin{itemize}
   \item[\dag] Este el primer item
    del tercer nivel
    \item[\dag] Este el segundo item
    del tercer nivel
    \item[\dag] Este el tercer item
    del tercer nivel
    \end{itemize}
 \end{itemize}

\end{itemize}
```

Pero tampoco está de más decir que los entornos son compatibles y que podemos insertar unos en otros, por lo tanto veamos un ejemplo en que metemos un entorno “itemize” dentro de uno “enumerate”.

```
\begin{enumerate}

\item Este es el primer caso,
No se me ocurre nada que decir,
pero como quiero mostrar un texto
largo\dots
 \begin{itemize}
 \item Este el primer item
 del segundo nivel
   \begin{itemize}
   \item Este el primer item
    del tercer nivel
    \item Este el segundo item
    del tercer nivel
    \item Este el tercer item
    del tercer nivel
    \end{itemize}
 \item Este el segundo item
 del segundo nivel
   \begin{itemize}
   \item Este el primer item
    del tercer nivel
    \item Este el segundo item
    del tercer nivel
    \item Este el tercer item
    del tercer nivel
    \end{itemize}
 \end{itemize}

\end{enumerate}
```

1. Este es el primer caso, No se me ocurre nada que decir, pero como quiero mostrar un texto largo...
    - Este el primer item del segundo nivel
        - Este el primer item del tercer nivel
        - Este el segundo item del tercer nivel
        - Este el tercer item del tercer nivel
    - Este el segundo item del segundo nivel
        - Este el primer item del tercer nivel
        - Este el segundo item del tercer nivel
        - Este el tercer item del tercer nivel

4.8 list

Es un entorno especial con el que se puede controlar hasta el último detalle del formato de una lista. La sintaxis es un poco más complicada:

`\begin{list}{`*etiqueta por defecto*`}{`*comandos*`}`

Se compone de tres partes. La declaración del entorno

`\begin{list}`

la declaración de la etiqueta por defecto del ítem, es decir, qué símbolo debe llevar cualquier ítem que no lo lleve explícito:

`{`*etiqueta por defecto*`}`

En cuanto a las etiquetas por defecto podemos usar cualquier signo o símbolo que se nos antoje, por ejemplo:

```
$\Box$
$\bullet$
$\rightarrow$
```

para colocar los símbolos:

□ • →

Los comandos se usan para establecer el formato de la lista:

`{`*comandos*`}`

Los que pueden usarse son los siguientes:

- `\topsep` El espacio vertical que quedará entre la lista y el texto normal que la rodea. Por defecto es 0.125in (pulgadas)

- `\partopsep` Espacio vertical extra que quedará entre la lista y el texto normal que la rodea, si éste es una línea en blanco.

- `\itemsep` La separación entre cada `\item`. Por defecto: 0.06in.

- `\parsep` Espacio vertical entre los distintos párrafos de un mismo `\item`. Por defecto 0.06in.

- `\leftmargin` Margen izquierdo del texto de un `\item` (sin incluir la etiqueta). El cálculo es relativo al margen izquierdo del texto. Por defecto: 2.5em (espacios tipo *em*).

- `\rightmargin` Margen derecho del texto de un `\item` (sin incluir la etiqueta). El cálculo es relativo al margen derecho del texto. Por defecto: 0em (espacios tipo *em*).

- `\listparindent` Sangría para todos los párrafos de todos los `\item`. Por defecto: 0in

- `\itemindent` Sangría delante de cada etiqueta. Por defecto: 0in (puede ser negativa).

- `\labelsep` Espacio horizontal entre la etiqueta y el texto de los `\item`. Por defecto: 0.5em

- `\labelwidth` Ancho de las etiquetas. Por defecto: 2em.

```
\begin{list}{$\Box$}{\itemsep .2cm
\parsep 0in \leftmargin 5em \labelsep 1em}

\item Este es el primer caso,
No se me ocurre nada que decir,
pero como quiero mostrar un
texto largo\dots

\item Este es el segundo caso.
Sigue sin salirme nada que decir,
pero insisto en mostrar un texto
largo\dots

\item Este es el tercer caso.
Pues no hay manera, no se me ocurre nada,
pero me obceco en mostrar un texto
largo\dots

\end{list}
```

- □ Este es el primer caso, No se me ocurre nada que decir, pero como quiero mostrar un texto largo...
- □ Este es el segundo caso. Sigue sin salirme nada que decir, pero insisto en mostrar un texto largo...
- □ Este es el tercer caso. Pues no hay manera, no se me ocurre nada, pero me obceco en mostrar un texto largo...

4.9 quotation

Se usa para realizar citas largas (con más de un párrafo) sangradas del texto principal. La única diferencia con `quote` (vid.

```
Y citemos una vez mas los famosos
versos del Cantar de los cantares
de Salomon:

\begin{quotation}
Cantar de los cantares de Salamon.
Besseme de los bessos de su boca,
ca mejores son tus amores que el vino.
...
Adonaronse tus mexillas con los
çarçillos, tu garganta con las sartas.
Çarçillos de oro te fare con granos
de plata.
\end{quotation}


Que podemos observar son bellos, hermosos,
como un reportaje de Felix Rodriguez.
```

Y citemos una vez más los famosos versos del Cantar de los cantares de Salomón:

> ≪Cantar de los cantares de Salamón. Bésseme de los bessos de su boca, ca mejores son tus amores que el vino. Al olor de tus ungüentos buenos, ungüento vaziadizo es el tu nonbre, por esso las donzellas te amaron.
>
> Atraeme; tras ty correré; metióme el rey en sus çilleros. Alegrar me hé e gozaré contigo, oleré tus amistades más que el vino; con derechumbres te amaron.Negra so yo e donosa, fijas de Iherusalem, como las tiendas de Quedar e como las cortinas de Salamón. Non me tachedes por que so baça, ca me priso el sol; los fijos de mi madre yraron contra mí; pusieron me guardadera delas viñas; mi viña que es mía non guardé.≫

Que podemos observar son bellos, hermosos, como un reportaje de Félix Rodríguez de la Fuente.

4.10 quote

Se usa para realizar citas más cortas que con el anterior (de un párrafo), pero funciona igual (excepto en la indentación del párrafo):

```
Y citemos una vez mas los famosos
versos del Cantar de los cantares
de Salomon:

\begin{quote}
Cantar de los cantares de Salamon.
Besseme de los bessos de su boca,
ca mejores son tus amores que el vino.
Al olor de tus ungüentos buenos,
ungüento vaziadizo es el tu nonbre,
por esso las donzellas te amaron.
\end{quote}

Que podemos observar son bellos, hermosos,
como un reportaje de Felix Rodriguez.
```

Y citemos una vez más los famosos versos del Cantar de los cantares de Salomón:

> ≪Cantar de los cantares de Salamón. Bésseme de los bessos de su boca, ca mejores son tus amores que el vino. Al olor de tus ungüentos buenos, ungüento vaziadizo es el tu nonbre, por esso las donzellas te amaron.≫

Que podemos observar son bellos, hermosos, como un reportaje de Félix Rodríguez de la Fuente.

4.11 tabbing

Se usa para hacer tablas muy sencillas (para tablas complejas vid.

```
\=
\kill
\>
\\
```

El primero sirve para fijar la posición de los tabuladores en la página. El segundo indica que se ha terminado de fijar tabuladores. El tercero, sería el tabulador en sí. El cuarto indica que se han terminado una fila. Veamos un ejemplo, el input:

```
\begin{tabbing}

\hspace*{.25in} \= \hspace*{1in} \= \hspace*{1in} \= \hspace*{1in} \kill

fila            \> col. 1       \> col. 2         \> col. 3 \\[.2in]
fila            \> col. 1       \> col. 2         \> col. 3 \\
fila            \> col. 1       \> col. 2         \> col. 3 \\
fila            \> col. 1       \> col. 2         \> col. 3 \\
fila            \> col. 1       \> col. 2         \> col. 3 \\
fila            \> col. 1       \> col. 2         \> col. 3 \\
fila            \> col. 1       \> col. 2         \> col. 3 \\

\end{tabbing}
```

produce el output:

fila col. 1 col. 2 col. 3

fila col. 1 col. 2 col. 3
fila col. 1 col. 2 col. 3
fila col. 1 col. 2 col. 3
fila col. 1 col. 2 col. 3
fila col. 1 col. 2 col. 3
fila col. 1 col. 2 col. 3

Lo bueno de este entorno es que permite alinear texto en columnas de tamaño fijo, y que las filas se puedan separar de una página a otra. El entorno `tabular`, sin embargo, no permite romper la tabla en varias páginas.

4.12 tabular

Es el hermano mayor del entorno anterior: la anchura de las columnas se adapta al texto más largo metido en dicha columna. Se pueden introducir líneas horizontales y verticales, y definir de antemano si el texto de una columna va centrado, a la izquierda o a la derecha... y muchas cosas más, difíciles de explicar y que hay que ver en ejemplos.

Este entorno tiene la forma:

`\begin{tabular}{`*formato*`}`

En donde `{`*formato*`}` define el número de columnas, si el texto de cada una va centrado o justificado (derecha o izquierda), si hay líneas verticales entre algunas columnas, etc.

La forma que toma es, por ejemplo, la siguiente:

```
\begin{tabular}{llc|cr|r}
```

es decir, `l` significa *izquierda*, `c` *centrado*, `r` *derecha*, y `|` que indica las líneas verticales entre las columnas. Habrá tantas columnas como `l, r, c` se especifiquen. Por lo tanto el ejemplo anterior especifica que tendremos 6 columnas: las dos primeras con el texto justificado a la izquierda, la tercera con texto centrado (una línea vertical), la cuarta centrada, la quinta a la derecha (una línea vertical) y la sexta con texto a la derecha.

Pero hay ejemplos más complejos que suponen el uso de otras variantes de formato, por ejemplo:

```
\begin{tabular}{|l p{4.5cm} rr|}
```

Es decir, cuatro columnas con dos líneas verticales (una a la izquierda de la tabla y la otra a la derecha). La segunda columna (`p{4.5cm}`) será de tipo párrafo con una anchura de 4.5cm.

```
\begin{tabular}{c r@{--}l}
```

Es decir, tres columnas: la primera centrada, la segunda con el texto a la derecha. Entre la segunda y la tercera (`{r@{--}l}`) se colocan los caracteres `--`. La arroba `@` indica que el espacio normal entre esas columnas queda anulado.

Veamos un ejemplo sencillo de cuatro columnas (las dos primeras con texto a la izquierda, la dos últimas con texto centrado), con líneas verticales entre todas las columnas. El comando `\hline` detrás de la primera fila indica una línea horizontal separándola de las demás.

```
\begin{tabular}{|l|l|c|c|}
\hline
Cuad.   & \hskip0.15in Folio & Total & Signat. \rule{0in}{3ex} \\[1ex]
\hline
16$^o$  & 117r-124v          & 8  &                  \\
17$^o$  & 125r-132v          & 8  &                  \\
18$^o$  & 133r-140v          & 8  & b.xviii          \\
19$^o$  & 141r-148v          & 8  &                  \\
20$^o$  & 149r-156v          & 8  & d.xx             \\
21$^o$  & 157r-164v          & 8  & a.xxi            \\
22$^o$  & 165r-172v          & 8  &                  \\
23$^o$  & 173r-180v          & 8  &                  \\
24$^o$  & 181r-188v          & 8  &                  \\
25$^o$  & 189r-196v          & 8  & a.ii-d.ii        \\
26$^o$  & 197r-204v          & 8  & a.iii-d.iii      \\
27$^o$  & 205r-212v          & 8  & b.iiii-c.iiii    \\
28$^o$  & 213r-220v          & 8  & b.v              \\
29$^o$  & 221r-225v          & *5 & c.               \\
        &                    &    &                  \\[1ex]
\hline
\end{tabular}
```

Cuad.	Folio	Total	Signat.
16^o	117r-124v	8	
17^o	125r-132v	8	
18^o	133r-140v	8	b.xviii
19^o	141r-148v	8	
20^o	149r-156v	8	d.xx
21^o	157r-164v	8	a.xxi
22^o	165r-172v	8	
23^o	173r-180v	8	
24^o	181r-188v	8	
25^o	189r-196v	8	a.ii-d.ii
26^o	197r-204v	8	a.iii-d.iii
27^o	205r-212v	8	b.iiii-c.iiii
28^o	213r-220v	8	b.v
29^o	221r-225v	*5	c.

Los comandos que se usan *dentro* del entorno `tabular` son los siguientes:

`&`	Separa las columnas.
`\\`	Indica fin de fila. Se puede añdir detrás `[`*espacio*`]`. En donde *espacio* es un espacio vertical añadido, por ejemplo, `[0.25in]`.
`\hline`	Dibuja una línea horizontal a través de todas las columnas.
`\vline`	Dibuja una línea vertical dentro de una fila.
`\cline{n-m}`	Dibuja una línea horizontal a través de las columnas n a m.
`\multicolumn{#1}{#2}{#3}`	Coloca un encabezamiento que se extiende sobre varias columnas. Toma tres argumentos: #1=nn (coloca el cabecero sobre las siguientes nn columnas. #2 define el formato del encabezamiento (por ejemplo `|c|`). #3 el texto del cabecero.
`\rule{0in}{`*espacio*`}`	Hace que la presente fila tenga una altura de *espacio* (por ejemplo `{3ex}`).

Veamos un ejemplo que usa todos esos comandos:

Como vamos a usar mucho `\rule{0in}{3ex}`, es bueno definir una macro para abreviar:

```
\newcommmand{\ESPACIO}{\rule{0in}{3ex}}
```

```
\newcommand{\ESPACIO}{\rule{0in}{3ex}}

\begin{tabular}{|l r@{. }l | p{1.0in}|}
%
\hline
%
\multicolumn{4}{|c|}
  {\bf Estructura del Poema de Mío Cid \ESPACIO}\\[1.5ex]
\multicolumn{4}{|c|}
  {\bf según Menéndez Pidal}\\[1.3ex]
%
\hline
\hline
%
\textit{Partes}\hspace{.2in}\vline & \multicolumn{2}{c}{versos} & \textit{explicación}
                                                              \ESPACIO \\[1ex]

Parte A\ESPACIO        &vv  & 0--150                  & suman 300\\
Parte B\ESPACIO        &vv  & 151--300                & versos\\ \cline{4-4}
Parte C\ESPACIO        &vv  & 300--1000               & suman 1700\\
Parte D\ESPACIO        &vv  & 1001--2000              & versos\\ \cline{4-4}
Parte E\ESPACIO        &vv  & 2000--2150              & suman 1000\\
Parte F\ESPACIO        &vv  & 2150--3000              & versos\\[1.3ex]
%
\hline
\end{tabular}
```

Estructura del Poema de Mío Cid según Menéndez Pidal

Partes	versos	*explicación*
Parte A	vv. 0–150	suman 300
Parte B	vv. 151–300	versos
Parte C	vv. 300–1000	suman 1700
Parte D	vv. 1001–2000	versos
Parte E	vv. 2000–2150	suman 1000
Parte F	vv. 2150–3000	versos

4.13 table

Este entorno lo único que hace es crear o actualizar un contador de tablas y permitir darle un título (mediante `\caption`) que luego irá al índice de tablas . Dentro de este entorno suelen colocarse los entornos que de hecho dibujan las tablas (`tabbing` y `tabular`), aunque no sea obligatorio. Si colocamos la tabla que hicimos en la sección anterior

```
\begin{table}
\caption{Ejemplo del entorno TABLE}

... la tabla anterior.

\end{table}
```

Tabla 1: Ejemplo del entorno TABLE

Estructura del Poema de Mío Cid según Menéndez Pidal		
Partes	versos	*explicación*
Parte A	vv. 0–150	suman 300
Parte B	vv. 151–300	versos
Parte C	vv. 300–1000	suman 1700
Parte D	vv. 1001–2000	versos
Parte E	vv. 2000–2150	suman 1000
Parte F	vv. 2150–3000	versos

4.14 verse

Sirve para destacar **poemas**. Obsérvese que el uso del indicador de fin de línea \\, es obligatorio para separar los versos. Las estrofas se separan con una línea en blanco:

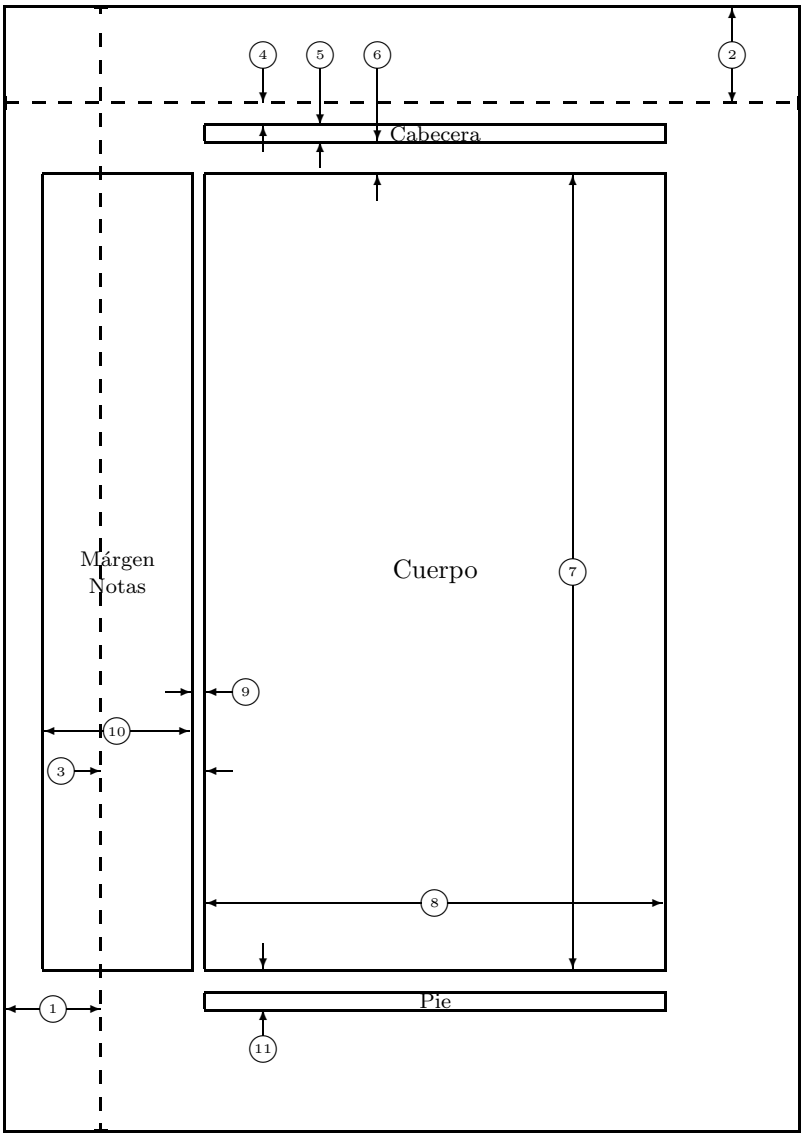
```
Observemos un poema
del Cantar de los Cantares
en la Biblia 10 288.

\begin{verse}
Cantar de los cantares de Salamon.\\
Besseme de los bessos de su boca,\\
ca mejores son tus amores que el vino.

Al olor de tus unguentos buenos,\\
unguento vaziadizo es el tu nonbre,\\
por esso las donzellas te amaron.
\end{verse}
```

Observemos un poema del Cantar de los Cantares en la Biblia 10 288.

Cantar de los cantares de Salamón.
Bésseme de los bessos de su boca,
ca mejores son tus amores que el vino.
Al olor de tus ungüentos buenos,
ungüento vaziadizo es el tu nonbre,
por esso las donzellas te amaron.

5 Diseño de la página

Figura 4: Parámetros del diseño de página



1	una pulgada + \hoffset	2	una pulgada + \voffset
3	\evensidemargin = 79pt	4	\topmargin = 17pt
5	\headheight = 12pt	6	\headsep = 25pt
7	\textheight = 598pt	8	\textwidth = 345pt
9	\marginparsep = 11pt	10	\marginparwidth = 111pt
11	\footskip = 30pt		\marginparpush = 5pt (no se muestra)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

6 Matemáticas

6.1 Símbolos matemáticos

He aquí unas tablas con unos cuantos de los símbolos usados en matemáticas:

Tabla 2: Acentos en modo matemático

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

Tabla 3: Relaciones binarias

Se pueden obtener las negaciones de estos añadiendo el comando `\not` como prefijo de los siguientes símbolos:

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` or `\le`	$\geq$	`\geq` or `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$ [a]	`\sqsubset` [a]	$\sqsupset$ [a]	`\sqsupset` [a]	$\bowtie$ [a]	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a] Hay que usar el paquete latexsym para acceder a este símbolo

Tabla 4: Operadores binarios

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$*$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$ [a]	`\lhd` [a]	$\rhd$ [a]	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$ [a]	`\unlhd` [a]	$\unrhd$ [a]	`\unrhd` [a]	$\wr$	`\wr`

Tabla 5: Operadores grandes

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

Tabla 6: Flechas

$\leftarrow$	`\leftarrow` or `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` or `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\iff$	`\iff` (bigger spaces)	$\leadsto$ [a]	`\leadsto` [a]

[a] Hay que usar el paquete latexsym para accder a este símbolo

Tabla 7: Delimitadores

(	(	)	)	$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`
[	[or `\lbrack`	]	] or `\rbrack`	$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`
{	`\{` or `\lbrace`	}	`\}` or `\rbrace`	$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`
$\langle$	`\langle`	$\rangle$	`\rangle`	|	| or `\vert`	$\|$	`|` or `\Vert`
$\lfloor$	`\lfloor`	$\rfloor$	`\rfloor`	$\lceil$	`\lceil`	$\rceil$	`\rceil`
/	/	\	`\backslash`		. (dual. empty)		

Tabla 8: Delimitadores largos

$\lgroup$	`\lgroup`	$\rgroup$	`\rgroup`	$\lmoustache$	`\lmoustache`	$\rmoustache$	`\rmoustache`
$\arrowvert$	`\arrowvert`	$\Arrowvert$	`\Arrowvert`	$\bracevert$	`\bracevert`		

Tabla 9: Símbolos misceláneos

$\dots$	`\dots`	$\cdots$	`\cdots`	$\vdots$	`\vdots`	$\ddots$	`\ddots`
$\hbar$	`\hbar`	$\imath$	`\imath`	$\jmath$	`\jmath`	ℓ	`\ell`
$\Re$	`\Re`	$\Im$	`\Im`	$\aleph$	`\aleph`	$\wp$	`\wp`
$\forall$	`\forall`	$\exists$	`\exists`	$\mho$ [a]	`\mho` [a]	∂	`\partial`
$'$	'	$\prime$	`\prime`	$\emptyset$	`\emptyset`	∞	`\infty`
∇	`\nabla`	$\triangle$	`\triangle`	$\Box$ [a]	`\Box` [a]	$\Diamond$ [a]	`\Diamond` [a]
$\bot$	`\bot`	$\top$	`\top`	$\angle$	`\angle`	$\surd$	`\surd`
$\diamondsuit$	`\diamondsuit`	$\heartsuit$	`\heartsuit`	$\clubsuit$	`\clubsuit`	$\spadesuit$	`\spadesuit`
$\neg$	`\neg` or `\lnot`	$\flat$	`\flat`	$\natural$	`\natural`	$\sharp$	`\sharp`

[a] Hay que usar el paquete latexsym para acceder a este símbolo