

# Apuntes de Latex

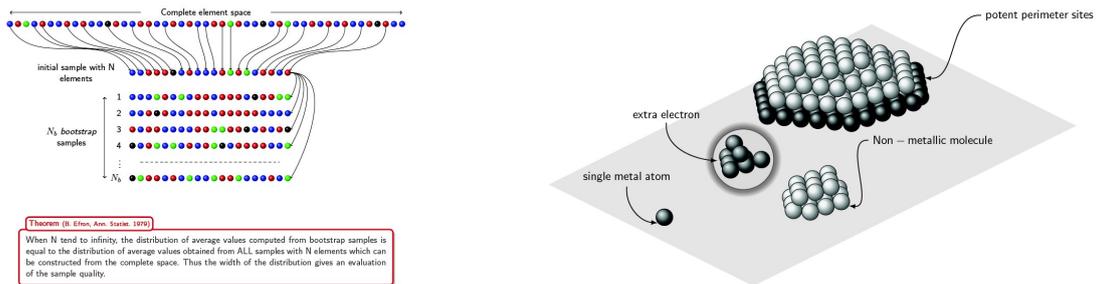
## Capítulo 1

### SECCIÓN 1

## Qué es L<sup>A</sup>T<sub>E</sub>X? Funcionamiento básico

L<sup>A</sup>T<sub>E</sub>X es un sistema avanzado de composición de textos, conocido popularmente por su amplio potencial para el proceso de textos científicos. Pero sus capacidades van mucho más allá de la escritura de fórmulas matemáticas (para lo cual funciona excelentemente –ver capítulo 3–). L<sup>A</sup>T<sub>E</sub>X posee amplias capacidades a la hora de configurar la apariencia general del documento (márgenes, cabeceras, división en secciones...). Asimismo, ofrece multitud de herramientas para la elaboración avanzada de índices, referencias cruzadas y bibliografía. El usuario tiene también a su disposición multitud de herramientas de maquetación para componer tanto párrafos de texto como gráficos, de forma absolutamente libre.

### Ejemplos de uso de recursos L<sup>A</sup>T<sub>E</sub>X



### Recursos gráficos con PGF

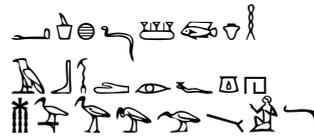
$$\int_D |\bar{\partial}u|^2 \Phi_0(z) e^{\alpha|z|^2} \quad (1)$$

$$\lim_{h \rightarrow +\infty} \int_{\Omega} |\nabla u_h| dx = |Du|(\Omega) \quad (2)$$

$$P_{r-j} = \begin{cases} 0 & r-j \text{ impar,} \\ r!(-1)^{(r-j)/2} & r-j \text{ par.} \end{cases} \quad (3)$$

W

er reitet so spät durch Wind?  
 Es ist der Vater mit Kind  
 Er hat den Knaben in Arm



### Fórmulas

### Efectos de texto



### Partituras musicales con MusiXTeX

Amplias capacidades gráficas están disponibles, mediante el empleo de paquetes adicionales (*pstricks*, *pgf*, *tikz*). Además de documentos impresos, existen utilidades como *beamer* para la creación de presentaciones de alta calidad. Finalmente, el carácter de código abierto del sistema y su forma modular hace posible el utilizar multitud de recursos programados por la comunidad de usuarios de  $\text{\LaTeX}$  para las aplicaciones más diversas, desde escritura de partituras musicales a diagramas de circuitos electrónicos.

El sistema  $\text{\TeX}/\text{\LaTeX}$ , a diferencia de procesadores de texto como MS-Word, no posee una interfaz gráfica interactiva en la cual según se compone el texto se observa directamente el resultado (lo que se conoce como editor tipo “WYSIWYG” ó WHAT-YOU-SEE-IS-WHAT-YOU-GET). En su lugar,  $\text{\TeX}/\text{\LaTeX}$  trabaja de forma similar a un lenguaje de programación<sup>1</sup>, compilando un fichero fuente (con la extensión *.tex*) del cual se obtiene como resultado un fichero procesado que podremos visualizar de diversas formas.

Los ficheros fuente *.tex* son simples archivos de texto ascii que pueden ser editados con cualquier editor de textos (aunque se sugiere trabajar desde entornos integrados como WinEdt, Texniccenter ó Kile<sup>2</sup>, ésto no es imprescindible) los cuales contienen tanto el texto en sí que queremos procesar, como comandos  $\text{\LaTeX}$  que se ocupan de formatear el texto. Todos estos comandos tienen en general (hay excepciones) la sintaxis:

$$\backslash\text{NombreComando}[\text{opciones}]\{\text{argumento}\}$$

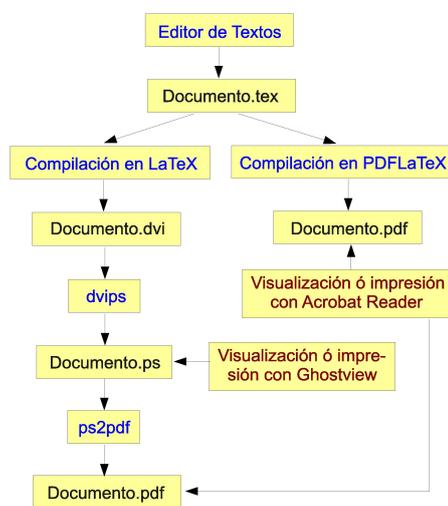
donde *NombreComando* es el nombre del comando en cuestión, *argumento* representa texto o variables  $\text{\LaTeX}$  sobre las que actúa el comando, y *opciones* denotan en general variables optativas que podemos ajustar. Es muy importante tener en cuenta que el nombre del comando es sensible a mayúsculas y minúsculas, por lo que debe escribirse siempre tal y como lo encontremos en la documentación  $\text{\LaTeX}$ . Además, no se permiten espacios entre el nombre del comando y su argumento (una fuente común de errores de sintaxis). El carácter  $\backslash$  tiene siempre la misión de señalar al compilador el comienzo de una instrucción.

Es esencial conocer que, a la hora de compilar un documento  $\text{\LaTeX}$ , existen dos posibilidades (ver gráfico adjunto):

- Compilar con el programa tradicional  $\text{\LaTeX}$ , lo cual da como resultado la creación de un fichero intermedio *Documento.dvi*, que contiene toda la información de formateado del documento original. Posteriormente, podemos transformar con el programa *dvips* nuestro archivo *.dvi* en un documento Postscript, el cual puede visualizarse con **Ghostview** ó imprimirse en una impresora postscript. Finalmente, la utilidad *ps2pdf* permite traducir documentos postscript al formato de acrobat **PDF**.

<sup>1</sup>De hecho,  $\text{\TeX}$  **ES** un lenguaje de programación; a lo largo del curso se explicarán algunos fundamentos de programación en  $\text{\TeX}$

<sup>2</sup>[www.winedt.com](http://www.winedt.com), [www.texniccenter.org](http://www.texniccenter.org), [kile.sourceforge.net](http://kile.sourceforge.net)



Funcionamiento de  $\text{\LaTeX}$

- Compilar con el relativamente nuevo programa PDF $\LaTeX$ , que permite obtener directamente como resultado de la compilación el documento en formato **PDF**.

Aunque ambas posibilidades pueden producir en numerosos casos el mismo resultado final, es crucial mencionar que **no son equivalentes**; el carácter más moderno del compilador PDF $\LaTeX$  implica que algunos recursos  $\LaTeX$  más antiguos pueden no estar disponibles, ó dar resultados erróneos. Por ejemplo, el conjunto de utilidades gráficas PSTricks, que hace un uso intensivo del lenguaje postscript, no soporta el uso de PDF $\LaTeX$  por lo que cualquier documento que contenga tales recursos ha de compilarse necesariamente con la secuencia  $\LaTeX$  + dvips + ps2pdf. Por contra, el paquete **beamer** para presentaciones está exclusivamente diseñado para trabajar en PDF $\LaTeX$ . *Se indicará, según vayan surgiendo estos casos especiales, cuál de las dos rutas debe emplearse.*<sup>3</sup>

### Ejemplo de documento fuente $\LaTeX$ y su de resultado compilado

```
\documentclass[a4,11pt]{article}
\usepackage[latini]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[spanish]{babel}
\renewcommand{\shorthandsspanish}{}

\title{Documento Fuente \LaTeX{}}
\author{Perico de los Palotes}
\date{}

\begin{document}

\maketitle
\tableofcontents

Eejmplo de documento \LaTeX\ de la clase
{\ttfamily article} con una estructura
reducida. Ésta incluye secciones,
subsecciones y una referencia cruzada.

\section{Primera sección}\label{primera}
Una primera sección con una fórmula y una lista.

\subsection{Fórmula}
Una ecuación: \(\frac{\partial f}{\partial x} =
\frac{\partial f}{\partial y}\)

\subsection{Listas}

Una lista de ítems señalados con una marca:
\begin{itemize}\itemsep=0pt
\item Primer ítem
\item Segundo ítem
\item Tercer ítem
\end{itemize}

\section{Segunda sección}
Ésta sección complementa a la sección \ref{primera}
incluyendo ejemplos de tablas escritas en \LaTeX.

\end{document}
```

#### Documento Fuente $\LaTeX$

Perico de los Palotes

#### Índice

1. Primera sección	1
1.1. Fórmula	1
1.2. Listas	1
2. Segunda sección	1
Eejmplo de documento $\LaTeX$ de la clase <code>article</code> con una estructura reducida. Ésta incluye secciones, subsecciones y una referencia cruzada.	

#### 1. Primera sección

Una primera sección con una fórmula y una lista.

##### 1.1. Fórmula

Una ecuación:  $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y}$

##### 1.2. Listas

Una lista de ítems señalados con una marca:

- Primer ítem
- Segundo ítem
- Tercer ítem

#### 2. Segunda sección

Ésta sección complementa a la sección 1 incluyendo ejemplos de tablas escritas en  $\LaTeX$ .

<sup>3</sup>Los detalles de cómo utilizar el entorno integrado WinEdt para compilar documentos se encuentran explicados en el archivo "WinEdt Minitutorial"

## SECCIÓN 2

## Conceptos básicos de formateado de texto

Todo documento L<sup>A</sup>T<sub>E</sub>X posee la siguiente estructura:

- **Preámbulo:** Declaraciones de carácter GLOBAL que afectan a la totalidad del documento
  - `\documentclass[opciones]{tipo_de_documento}` → **OBLIGATORIA**; éste debe de ser además el primer comando del documento. Mediante ésta declaración indicamos a L<sup>A</sup>T<sub>E</sub>X que tipo de documento (book, article, report, letter...) queremos escribir
  - `\usepackage[opciones]{paquete}` → carga de paquetes con utilidades (para incluir gráficos, texto en color, presentaciones, etc...)
  - Otras declaraciones: Interlínea, formato de página, fuentes, etc...
- **Cuerpo:** Todo lo comprendido entre `\begin{document}` y `\end{document}`, es decir, el documento propiamente dicho.

En la página anterior puede verse un ejemplo de documento simple, tomando la forma de la clase `article`.

### 2.1. Reglas generales de composición de texto

Existen ciertas reglas generales a la hora de escribir texto:

- **Texto alineado y centrado:** El texto se alinea y justifica automáticamente, según medidas predeterminadas o impuestas por nosotros; **IMPORTANTE:** Toda medida predeterminada es ajustable
- **Los espacios se ignoran:** Da igual separar las palabras con 1 espacio o varios. Para aumentar el espacio de separación entre palabras se usa: `\` seguido de espacio
- **Punto y aparte:** Dejar una línea en blanco equivale a cambiar de párrafo (punto y aparte). Lo mismo se obtiene con `\par`. **Ojo!** Nótese que dejar *varias* líneas en blanco es igual a dejar una: el efecto es el mismo, i.e., comienzo de un nuevo párrafo.
- **Cambio de línea:** Se puede cambiar de línea, *sin cambiar de párrafo*, usando `\\`

Los ejemplos en la página siguiente ilustran éstos puntos. Es importante tener en cuenta que L<sup>A</sup>T<sub>E</sub>X, por defecto, coloca una pequeña indentación al comienzo de cada párrafo. Para controlar ésta indentación se dispone del comando `\parindent=Xmm` (más adelante se explica el manejo de unidades de longitud). Éste comando cambia la indentación de párrafo de forma *global*, afectando a todos los párrafos tras el comando. Si se desea suprimir la indentación para un párrafo en particular, puede hacerse colocando `\noindent` al comienzo del mismo. El espaciado estándar entre párrafos (nulo por defecto) se puede modificar a través del comando `\parskip=Xmm`

### Manejo de espacios, cambios de línea y de párrafo

En un lugar de la       mancha  
de cuyo nombre no quiero  
acordarme, \ \ \ \ no ha mucho tiempo  
que \ \ vivía un hidalgo de los de  
lanza en astillero, adarga antigua,  
rocín flaco y galgo corredor. \par Una  
olla de algo más vaca que carnero,  
      salpicón las más noches, duelos y  
quebrantos los sábados, lentejas  
los viernes, algún  
palomino de añadidura los domingos,  
consumían las tres partes de su hacienda.

En un lugar de la mancha de cuyo nombre  
no quiero acordarme,       no ha mucho tiempo  
que  
vivía un hidalgo de los de lanza en astillero,  
adarga antigua, rocín flaco y galgo corredor.  
Una olla de algo más vaca que carnero, sal-  
picón las más noches, duelos y quebrantos los  
sábados, lentejas los viernes, algún  
      palomino de añadidura los domingos, consu-  
mían las tres partes de su hacienda.

### Control de la indentación y del espaciado entre párrafos

\parindent=8mm  
En un lugar de la mancha de cuyo nombre no  
quiero acordarme, no ha mucho tiempo que  
vivía un hidalgo de los de lanza en astillero,  
adarga antigua, rocín flaco y galgo corredor.

Una olla de algo más vaca que carnero,  
salpicón las más noches, duelos y quebrantos  
los sábados, lentejas los viernes, algún  
palomino de añadidura los domingos,  
consumían las tres partes de su hacienda.

\noindent  
En un lugar de la mancha de cuyo nombre no  
quiero acordarme, no ha mucho tiempo que  
vivía un hidalgo de los de lanza en astillero,  
adarga antigua, rocín flaco y galgo corredor.

\parskip=3mm  
Una olla de algo más vaca que carnero,  
salpicón las más noches, duelos y quebrantos  
los sábados, lentejas los viernes, algún  
palomino de añadidura los domingos,  
consumían las tres partes de su hacienda.

En un lugar de la mancha de cuyo nombre no  
quiero acordarme, no ha mucho tiempo que  
vivía un hidalgo de los de lanza en astillero,  
adarga antigua, rocín flaco y galgo corredor.

En un lugar de la mancha de cu-  
yo nombre no quiero acordarme, no ha  
mucho tiempo que vivía un hidalgo de  
los de lanza en astillero, adarga anti-  
gua, rocín flaco y galgo corredor.

Una olla de algo más vaca que car-  
nero, salpicón las más noches, duelos  
y quebrantos los sábados, lentejas los  
viernes, algún palomino de añadidura  
los domingos, consumían las tres par-  
tes de su hacienda.

En un lugar de la mancha de cuyo nom-  
bre no quiero acordarme, no ha mucho  
tiempo que vivía un hidalgo de los de  
lanza en astillero, adarga antigua, rocín  
flaco y galgo corredor.

Una olla de algo más vaca que car-  
nero, salpicón las más noches, duelos  
y quebrantos los sábados, lentejas los  
viernes, algún palomino de añadidura  
los domingos, consumían las tres par-  
tes de su hacienda.

En un lugar de la mancha de cu-  
yo nombre no quiero acordarme, no ha  
mucho tiempo que vivía un hidalgo de  
los de lanza en astillero, adarga anti-  
gua, rocín flaco y galgo corredor.

## SECCIÓN 3

## Espacios horizontales y verticales

Un elemento muy importante a la hora de contruir un documento son las longitudes que se utilizan para delimitar diferentes distancias entre objetos. Existe una amplia variedad de *unidades de longitud* que podemos utilizar:

Unidades de longitud:

Medidas absolutas		
pt	punto	1pt $\approx$ 0.35146 mm
pc	pica	1pc = 12 pt
in	pulgada	1in = 72.27 pt = 2.54 cm
cm	centímetro	
mm	milímetro	1 mm = 2.845 pt
dd	didot	1157 dd = 1238 pt
cc	cícero	1 cc = 12 dd
Medidas relativas		
em	aprox. la anchura de una ‘M’ de la fuente en curso	
ex	aprox. la anchura de una ‘x’ de la fuente en curso	

Las medidas relativas “em” y “ex” dependen del tamaño de letra en curso que estemos utilizando. Existen muchos otros ejemplos de uso de medidas relativas; en general, L<sup>A</sup>T<sub>E</sub>X maneja internamente una amplia cantidad de longitudes que en principio desconocemos o que incluso son *elásticas*. Es un procedimiento frecuente, y bastante útil, el manejar longitudes relativas. Por ejemplo, si queremos delimitar la anchura de un objeto cualquiera como la mitad de la anchura del texto en la página, definida a través de la variable `\textwidth`, podemos utilizar la longitud relativa `0.5\textwidth`.

Para modificar el valor de medidas de longitud (como `\parskip`, `\parindent`, etc...) existen dos sintaxis de comando equivalentes:

`\Longitud=Xmm`                      `\Longitud Xmm`

Hay varias formas de añadir espacios verticales entre diversos objetos:

- `\\` y `\newline` → Análogos: salto de línea simple. En este punto es bueno recordar la diferencia con `\par`: éste último *cambia de párrafo*, mientras que los anteriores simplemente terminan bruscamente un línea y pasan a la siguiente, sin comenzar nuevo párrafo.
- `\\[Salto]` → Espacio vertical de longitud *Salto*
- `\vspace{Salto}` → Análogo al anterior; nótese que no tiene efecto *al comienzo de una página*, en tal circunstancia se puede usar el comando análogo `\vspace*{Salto}`, que evita este problema
- Espacios verticales de longitud predefinida:

- `\bigskip` → approx. 1 línea en blanco
- `\medskip` → approx. 1/2 del espacio de una línea en blanco
- `\smallskip` → approx. 1/4 del espacio de una línea en blanco

Para añadir espacios horizontales podemos utilizar:

- `\hspace{longitud}` → Espacio horizontal de extensión *longitud*
- `\hspace*{longitud}` → Igual, válido *al comienzo de una línea*
- Predefinidos:
  - `\_` → un espacio entre palabras
  - `\enskip` → medio “em”
  - `\quad` → un “em”
  - `\qquad` → dos “em”

Si queremos saltar de página, podemos utilizar tanto `\newpage` como `\clearpage`, que indican a L<sup>A</sup>T<sub>E</sub>X que se debe finalizar la página en curso y comenzar una nueva. Nótese que `\clearpage` posee un significado especial: en el caso de que haya *elementos flotantes* (como figuras y tablas, para los cuales L<sup>A</sup>T<sub>E</sub>X se encarga por si solo de buscar la ubicación más adecuada) pendientes de ubicar, tales elementos se imprimirán inmediatamente en la página siguiente, formada por sólo tablas y gráficas. El texto se reiniciará entonces otra página después.

La distancia entre líneas (interlínea) también es modificable, situando en el preámbulo (**OJO!**, y sólo en el preámbulo, fuera de ahí la instrucción no funciona) la instrucción:

```
\renewcommand*{\baselinestretch}{Número}
```

lo que **escala** la interlínea por la cantidad Número (un valor de 2.0 equivaldría a doble espaciado). Si se quiere cambiar el valor de la interlínea en distintas partes del documento, se puede utilizar el paquete `setspace`, con la sintaxis:

```
\usepackage[espaciado]{setspace}
```

lo cual hace el valor de la interlínea en todo el documento igual a `espaciado`. Los posibles valores son: `singlespacing`, `onehalfspacing` y `doublespacing`, que equivalen respectivamente a un valor de `baselinestretch` de 1, 1.5 y 2 (si no se incluye la opción `espaciado`, el valor por defecto es `singlespacing`). Después, en el cuerpo del documento, puede variarse a voluntad la interlínea mediante los comandos: `\singlespacing`, `\onehalfspacing` y `\doublespacing`.

#### SECCIÓN 4

## Centrado y justificación a los márgenes del texto

Un texto dado puede ajustarse a cualquiera de los dos lados de la página ó al centro:

- Para centrar un párrafo se utiliza el *entorno* `center`

```
\begin{center}
Texto a centrar
\end{center}
```

Ejemplo:

```
\begin{center}
El ingenioso hidalgo\
D. Quijote de la Mancha\|[0.3cm]
Miguel de Cervantes Saavedra
\end{center}
```

```
El ingenioso hidalgo
D. Quijote de la Mancha
Miguel de Cervantes Saavedra
```

- Para alinear a los lados, tenemos los entornos **flushleft** y **flushright**

```
\begin{flushleft}
Probando \
la forma de alinear \
por la izquierda
\end{flushleft}
```

```
\begin{flushright}
Probando \
la forma de alinear \
por la derecha
\end{flushright}
```

---

Probando  
la forma de alinear  
por la *izquierda*

Probando  
la forma de alinear  
por la *derecha*

- Para textos pequeños, menores que una línea, se pueden utilizar, respectivamente:

```
\leftline{Texto}
```

```
\centerline{Texto}
```

```
\rightline{Texto}
```

Es interesante el hecho de que existen en L<sup>A</sup>T<sub>E</sub>X multitud de comandos que poseen ambas versiones, una “corta”, de tipo:

```
\comando{argumento}
```

donde el comando afecta a un texto pequeño (argumento), y otra “larga”, denominada **entorno** de tipo:

```
\begin{entorno} Texto largo \end{entorno}
```

Alternativamente, también existe otra método para éste último procedimiento, incluyendo el comando dentro de un **grupo**:

```
{\comando Objeto extenso}
```

donde los *delimitadores* { y } definen el grupo de objetos a los que afectará el comando.

## SECCIÓN 5

## Silabeo

Puede ocurrir que los algoritmos de silabeo de  $\text{\LaTeX}$  no funcionen correctamente y que al cambiar de línea se rompa una palabra de forma inadecuada. Para evitarlo hay dos alternativas:

- Utilizar la instrucción `\hyphenation{lista de palabras}` en el preámbulo; por ejemplo, `\hyphenation{For-tran fi-che-ro}` sólo permitirá la división de las palabras “fortran” y “fichero” por los lugares indicados. Nótese que no se permiten caracteres con acentos ó símbolos en el argumento, y que no se hacen distinciones entre las letras mayúsculas y minúsculas de las palabras en la orden.
- Fuera del preámbulo, se puede utilizar la instrucción: `\-` Ésta es válido utilizarla en palabras con acentos ó símbolos, por ejemplo: `te\-\lé\-\fo\-\no`

## SECCIÓN 6

## Escribiendo en castellano

El idioma por defecto de  $\text{\LaTeX}$  es el inglés. Esto quiero decir que, de forma estándar, no se reconocen los caracteres especiales como Ñ, letras acentuadas, etc... Además, definiciones por defecto, como títulos para capítulos, fechas, etc, estarán en inglés. Pruébese por ejemplo lo siguiente:

```
\documentclass{article}
\begin{document}
Saludos desde \LaTeX. Haciendo una compilación de prueba, de texto
en español, para ver si todo funciona.

Escrito y compilado el día \today.
\end{document}
```

y se verá que los acentos y ñ desaparecen y que la fecha (comando `\today`) se imprime en inglés. Para solucionar tales problemas se pueden incluir los siguientes paquetes en el preámbulo del documento:

- `\usepackage[latin1]{inputenc}` → Para que  $\text{\LaTeX}$  entienda los símbolos del teclado español. Con este paquete podemos teclear directamente símbolos del teclado que serán reconocidos por el compilador <sup>4</sup>
- `\usepackage[T1]{fontenc}` → Para que utilice nuestros tipos acentuados, en vez de construirlos con METAFONT

<sup>4</sup>a excepción del símbolo del euro (€); véase mas adelante

- `\usepackage[spanish]{babel}` → Reglas españolas para división de sílabas, traducción de comandos, etc...
- `\renewcommand{\shorthandsspanish}{}` → Desactiva métodos taquigráficos en español (que pueden molestartos)

Podemos utilizar las declaraciones anteriores como cabecera estándar para escribir textos en español. Llegado este punto, es bueno destacar que existen alternativas para escribir caracteres acentuados sin utilizar el teclado español y la codificación “latin1” en el paquete `inputnc`. Ésto puede ser útil si, por ejemplo, nos encontramos en el extranjero ó carecemos del teclado adecuado. Para ello, desactivaríamos las declaraciones `\usepackage[latin1]{inputenc}` y `\usepackage[T1]{fontenc}`, haciendo uso de las instrucciones expuestas en la siguiente tabla para conseguir caracteres acentuados:

ò	<code>\‘o</code>	ó	<code>\’o</code>	ô	<code>\^o</code>	õ	<code>\~o</code>
ō	<code>\=o</code>	ò	<code>\.o</code>	ö	<code>\"o</code>	ç	<code>\c c</code>
õ	<code>\u o</code>	ö	<code>\v o</code>	õ	<code>\H o</code>	ø	<code>\d o</code>
o	<code>\b o</code>	oo	<code>\t oo</code>				

**IMPORTANTE:** Las letras *i* y *j* necesitan un tratamiento especial, dado que no deben tener sus puntos antes de ser acentuadas. Su eliminación se consigue con los comandos `\i` y `\j`, respectivamente. Así, para obtener, por ejemplo:

Él está aquí

se debería escribir `\’{E}l est\’{a} aqu\’{\i}`

Para otros símbolos pertenecientes a diversos idiomas véase la siguiente tabla:

œ	<code>\oe</code>	Œ	<code>\OE</code>	æ	<code>\ae</code>	Æ	<code>\AE</code>
å	<code>\aa</code>	Å	<code>\AA</code>				
ø	<code>\o</code>	Ø	<code>\O</code>	ı	<code>\l</code>	Ł	<code>\L</code>
ß	<code>\ss</code>						
ı	<code>!‘</code>	ı	<code>?‘</code>				

## SECCIÓN 7

### Más sobre signos ortográficos

En esta sección describiremos diversos comandos de utilidad a la hora de escribir un texto (comillas, guiones, ordinales, etc...)

#### 7.1. Los diez caracteres reservados

En  $\text{\LaTeX}$  existen 10 caracteres especiales que el sistema utiliza para distintos propósitos, a saber:

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

los cuales tienen los siguientes usos:

$\$  Indicador de comando.

{ } Delimitadores de grupos.

# Nombra los argumentos de un comando.

& Separa columnas de una tabla.

% Se utiliza para introducir comentarios: En una línea del fichero fuente dada, todo lo que se encuentre a la derecha de éste signo es ignorado por el compilador y se entiende como comentario.

~ Se utiliza para evitar la separación de palabras: Es una conocida norma tipográfica el no separar términos complementarios, como por ejemplo Sr. Director ó A. Einstein. Utilizando la tilde como ligadura, se evita que L<sup>A</sup>T<sub>E</sub>X rompa éstas palabras en dos al cambiar de línea: Sr.~Director, A.~Einstein <sup>5</sup>

\$ \_ ^ Se utilizan en fórmulas matemáticas.

El carácter reservado significa que no podemos incluirlos en el texto normal simplemente escribiéndolos. En la siguiente tabla se muestra la sintaxis que se debe utilizar para escribirlos dentro de un documento:

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

## 7.2. Comillas, guiones, puntos suspensivos, grados, etc..

### Comillas:

Tecleamos: << Texto >>      Resulta: « Texto »

Tecleamos: ‘‘ Texto ’’      Resulta: “ Texto ”

Tecleamos: ‘ Texto ’      Resulta: ‘ Texto ’

### Guiones:

Tecleamos: -      Resulta: -

Tecleamos: --      Resulta: –

<sup>5</sup>Otro método es encerrar la frase dentro de un caja:  $\mbox{Texto}$

Tecleamos: --- Resulta: —

Tecleamos: \$-\$ Resulta: −  
(signo matemático menos)

### Puntos suspensivos:

Hay varias formas de introducir los puntos suspensivos:

- `...` → (la más sencilla)
- `\dots` → (sólo funciona con la opción `spanish` de `babel`)
- `\dots` → (comando propio de `LATEX`, siempre disponible)
- `\ldots` → Distancia entre puntos suspensivos algo mayor

### Ordinales y grados:

Para obtener ordinales abreviados (1<sup>a</sup> ó 1<sup>o</sup>), podemos hacerlo directamente desde el teclado. Para escribir otros ordinales, puede hacerse con:

`\textsuperscript{Superíndice}`

Por ejemplo, `3\textsuperscript{er}` produce 3<sup>er</sup>. Otra versión (sólo disponible en `babel`, versión `spanish`) es:

`\sptext{Superíndice}`

que introduce un punto antes del superíndice y cambia el tamaño del superíndice si es un carácter en mayúsculas: 2.<sup>A</sup>, 3.<sup>er</sup>. Para generar correctamente el signo de grado, se utiliza: `\textdegree`

### Otros signos:

El siguiente cuadro muestra como obtener otros signos diversos:

Comando	Resultado	Comando	Resultado
<code>\dag</code>	†	<code>\ddag</code>	‡
<code>\S</code>	§	<code>\P</code>	¶
<code>\textbullet</code>	•	<code>\textvisiblespace</code>	~
<code>\textregistered</code>	®	<code>\copyright</code>	©
<code>\texttrademark</code>	™	<code>\pounds</code>	£

Los comandos `\copyright` y `\textregistered` son casos particulares de un comando más general: `\textcircled{Carácter}`, que encierra `Carácter` dentro de un círculo.

## 7.3. El euro

Debido a la modernidad del símbolo del euro, la opción `latin1` del paquete `inputenc` desgraciadamente aún no reconoce éste signo. La solución para obtener el símbolo del euro está en cargar en el preámbulo el paquete `eurosym`:

```
\usepackage{eurosym}
```

tras lo cual se obtiene en símbolo € con el comando `\euro`. Puede incluso mejorarse la situación insertando la siguiente declaración en el preámbulo (por supuesto, **después** del comando `\usepackage{eurosym}`):

```
\DeclareInputText{128}{\euro}6
```

que asignaría el símbolo € del teclado la instrucción “`\euro`”, lo que ya permite utilizar el símbolo del teclado con normalidad.

---

SECCIÓN 8

---

## Párrafos especiales: quote, quotation, verse, y más...

Los entornos `quote` y `quotation` permiten introducir citas textuales, en párrafos ligeramente más pequeños que el texto base; por ejemplo:

La inclusión de citas textuales, como la del escritor Bertolt Brecht que viene a continuación, es una tarea sencilla con `\LaTeX`.

```
\begin{quote}
```

Hay personas que luchan un día, y son buenas. Hay otras que luchan un año y son mejores. Hay quienes luchan muchos años, y son muy buenas.

Pero hay algunas que luchan toda la vida: ésas son las imprescindibles

```
\end{quote}
```

produce:

Con `quote`:

La inclusión de citas textuales, como la del escritor Bertolt Brecht que viene a continuación, es una tarea sencilla con `LATEX`.

Hay personas que luchan un día, y son buenas. Hay otras que luchan un año y son mejores. Hay quienes luchan muchos años, y son muy buenas.

Pero hay algunas que luchan toda la vida: ésas son las imprescindibles

Con `quotation`:

La inclusión de citas textuales, como la del escritor Bertolt Brecht que viene a continuación, es una tarea sencilla con `LATEX`.

Hay personas que luchan un día, y son buenas. Hay otras que luchan un año y son mejores. Hay quienes luchan muchos años, y son muy buenas.

Pero hay algunas que luchan toda la vida: ésas son las imprescindibles

Como se puede ver, `quotation` introduce sangrado en los párrafos de la cita, y disminuye la separación entre los mismos.

Otro entorno predefinido es el `verse`, para escribir versos. Tiene la siguiente sintaxis:

---

<sup>6</sup>En Windows. Para Linux, el código del carácter € es 164

```

\begin{verse}
verso1 \\
verso2 \\
....
\end{verse}

```

Utilizando el paquete `shapepar` se pueden contruir párrafos con formas muy especiales. Así por ejemplo, empleando: `\heartpar{Texto del párrafo a formatear}` puede obtenerse lo siguiente:

El 30 de marzo de  
1977, el profesor Donald E. Knuth, de  
la Universidad de Stanford, recibió las galeradas  
o pruebas de imprenta de la segunda edición del segundo  
volumen de su famosa obra *The Art of Computer Programming*. La impresión que dichas pruebas causaron al autor fue nefasta; él mismo las calificó de tipográficamente horribles y tan importantes le parecieron los problemas a los que se enfrentaba que decidió resolverlos por sí mismo. A partir de las ideas de Gutenberg y utilizando las computadoras como herramientas, Knuth creó T<sub>E</sub>X, un sistema para escribir textos científicos (especialmente matemáticos), cómodo y transportable entre plataformas, que muchos consideran ahora como la aportación más importante realizada en este campo desde la imprenta de Gutenberg.

♡

Lo anterior es una aplicación del comando general `\parshape`, que permite construir párrafos de forma arbitraria. Su sintaxis es la siguiente:

```
\parshape=n i1 l1 i2 l2...in ln
```

e indica que las primeras  $n$  líneas del párrafo tendrán longitudes  $l_1, \dots, l_n$ , respectivamente, y estarán sangradas  $i_1, \dots, i_n$ , respectivamente. Si el párrafo tiene más de  $n$  líneas, las condiciones para la  $n$ -ésima serán repetidas hasta final de párrafo. Para cancelar el comando, basta incluir `\parshape=0`.

## SECCIÓN 9

## Tipos de letra

## 9.1. Familias

Por defecto, L<sup>A</sup>T<sub>E</sub>X utiliza los tipos Computer Modern Fonts, creados por D.E. Knuth para su utilización en T<sub>E</sub>X.<sup>7</sup> Estos tipos agrupan tres familias diferentes:

- Roman (la opción por defecto)
- Sanserif (sin adornos)
- Typewriter (tipo máquina de escribir)

con las siguientes instrucciones para obtenerlas, respectivamente:

- `\textrm{Texto}` (roman) `\rmfamily Texto`
- `\textsf{Texto}` (sanserif) `\sffamily Texto`
- `\texttt{Texto}` (typewriter) `\ttfamily Texto`

Los comandos a la izquierda corresponden al modo *Texto*, es decir, para textos cortos **no más largos que un párrafo**. Los comandos a la izquierda se mantienen hasta que se declare una nueva familia, aunque lo usual suele ser incluirlos dentro de un *grupo*, en la forma siguiente:

Esto es roman, `{\sffamily esto es sanserif}`,  
`{\ttfamily esto es typewriter}`, y esto sigue siendo roman.

que produce:

Esto es roman, esto es sanserif, esto es typewriter, y esto sigue siendo roman.

Completamente análogo a `{\sffamily Texto}` sería utilizar el siguiente entorno:

`\begin{sffamily} Texto extenso... \end{sffamily}`

## 9.2. Perfiles

Para cada familia tenemos cuatro posibles perfiles, recto (opción por defecto, *itálico*, *inclinado* (slanted) y VERSAL (letras mayúsculas pequeñas), cuyos comandos correspondientes son:

- `\textup{Texto}` (recto) `\upshape Texto`
- `\textit{Texto}` (*itálico*) `\itshape Texto`
- `\textsl{Texto}` (*inclinado*) `\slshape Texto`
- `\textsc{Texto}` (VERSAL) `\scshape Texto`

<sup>7</sup>Pueden cargarse otros muchos tipos, lo cual se verá mas adelante

### 9.3. Grosor

Finalmente, hay dos grosores (también llamados series) para cada tipo: el normal ó medio (opción por defecto) y el grueso ó negrita. Se activan con:

- `\textmd{Texto}` (medio) `\mdseries Texto`
- `\textbf{Texto}` (**grueso**) `\bfseries Texto`

Todas las características explicadas (familia, perfil y grosor) pueden combinarse (aunque puntualmente alguna opción mixta no esté disponible). Por ejemplo, `\bfseries\itshape` produce *letra negrita itálica*.

### 9.4. Enfatizar

Se puede resaltar texto con el comando `emph{Texto}` (ó `\em Texto`, en modo extendido) lo cual pone en itálica el texto si el ambiente es normal, ó pone normal el texto si el ambiente es itálico.

### 9.5. Tamaño

Tenemos a nuestra disposición los siguientes tamaños, los cuales son *relativos* a la fuente estándar del documento (que puede cambiarse, ya se verá mas adelante cómo)

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

## 9.6. Colores

El paquete `color` permite colorear un texto. Se puede cargar como `\usepackage{color}`, lo cual permite sólo usar unos pocos colores básicos (`white`, `black`, `red`, `blue`, `yellow`, `green`)<sup>8</sup>

Para cambiar el color, se usan los comandos:

- `\textcolor{NombreColor}{Texto}`
- `\color{NombreColor}`

siendo la primera la versión corta, y la segunda la versión larga que tendrá efecto hasta que se cambie de nuevo el color (ó hasta que termine el grupo, si se usa:  
`{\color{NombreColor} Texto extenso... }`

Todas las características de tipo, tamaño, forma, grosor, color, etc... para texto escrito pueden combinarse mediante la anidación de comandos, como muestra el siguiente ejemplo:

```
{\Large\bfseries\color{blue} Esto es letra grande, negrita y azul}
```

**Esto es letra grande, negrita y azul**

```
{\small\sffamily\itshape\color{red} Letra pequeña, sanserif, itálica y roja}
```

*Letra pequeña, sanserif, itálica y roja*

```
{\huge\textcolor{green}{\textbf{\textsc{Letra Mayúscula  
negrita, verde y muy grande}}}
```

**LETRA MAYÚSCULA NEGRITA, VERDE Y MUY GRANDE**

## 9.7. El paquete `soul`

Cargando el paquete `soul`, podemos incorporar los siguientes efectos de resaltado de texto:

```
\hl{Texto}
```

**Texto marcado** (para ésto hace falta tener también cargado el paquete `color`)

```
\ul{Texto}
```

En un lugar de la mancha de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.

---

<sup>8</sup>En un capítulo posterior se explicará como manejar cualquier tipo de color, además de la posibilidad de colorear páginas, cajas, etc...

`\st{Texto}`

En un lugar de la Mancha de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, ~~rocin flaco y galgo corredor~~. Una ~~olla de algo más vaca que carnero~~, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.

`\so{Texto}`

Texto resaltado (con `l e t r a s e s p a c i a d a s` más de lo normal)