

---

# Capítulo 1

## Cosas que debe saber

### 1.1. $\text{\LaTeX}$

$\text{\LaTeX}$  es un programa de computador creado por Donald E. Knuth. Sirve para componer texto y fórmulas matemáticas. Knuth empezó a escribir el motor de composición  $\text{\LaTeX}$  en 1977 para investigar el potencial de los equipos de impresión digital que estaban empezando a usarse en la industria de la impresión en aquel tiempo; en concreto tenía la esperanza de poder revertir la tendencia de calidad tipográfica en declive que él vio afectar a sus propios libros y artículos.

$\text{\LaTeX}$  se pronuncia “látej” o “latech”. La “j” surge del alfabeto griego donde X es la letra “j” o “ji”.  $\text{\LaTeX}$  es también la primera sílaba de la palabra griega  $\tau\epsilon\chi\eta\nu$  (arte).

### 1.2. Lo básico

#### 1.2.1. Autor, maquetador y compositor

Para publicar algo, los autores dan su manuscrito mecanografiado a una editorial. Uno de sus maquetadores decide el aspecto del documento (anchura de columna, tipografías, espacio ante y tras cabeceras, etc.). El maquetador escribe sus instrucciones en el manuscrito y luego se lo da al compositor o cajista, quien compone el libro siguiendo esas instrucciones.

Un maquetador humano suele interpretar qué pretendía el autor mientras escribía el manuscrito. Decide sobre las cabeceras de los capítulos, las citas, los ejemplos, las fórmulas, etc., basándose en su conocimiento profesional y en el contenido del manuscrito.

En un entorno  $\text{\LaTeX}$ ,  $\text{\LaTeX}$  representa el papel del maquetador. Pero  $\text{\LaTeX}$  es “sólo” un programa, y por tanto necesita más supervisión. El autor tiene que proporcionar información adicional para describir la estructura lógica de su trabajo. Tal información se escribe entre el texto como “órdenes  $\text{\LaTeX}$ ”.

Esto es bastante diferente del enfoque visual o WYSIWYG<sup>1</sup> que sigue la mayoría de los procesadores de texto modernos, como *Open Office Writer* o *Microsoft Word*. Con estos programas, los autores especifican el aspecto del documento interactivamente mientras escriben texto en el computador. Así pueden ver en la pantalla cómo aparecerá el trabajo final cuando se imprima.

Cuando use  $\text{\LaTeX}$  no suele ser posible ver el aspecto final del texto mientras lo escribe, pero tal aspecto puede verse en pantalla tras procesar el archivo mediante  $\text{\LaTeX}$ . Entonces pueden hacerse correcciones antes de enviar el documento a la impresora para tener una copia en papel.

### 1.2.2. Maquetación

La maquetación (diseño tipográfico) es un arte. Los autores sin habilidad a menudo cometen errores de formato al suponer que maquetar es mayormente una cuestión de estética, “Si un documento luce bien artísticamente, está bien diseñado”. Pero como un documento se escribe para ser leído y no colgado en una galería de arte, su legibilidad es mucho más importante que su aspecto. Ejemplos.

- El tamaño de los tipos y la numeración de las cabeceras debe escogerse para que la estructura de capítulos y secciones quede clara al lector.
- La longitud de línea debe ser suficientemente corta para no cansar a los ojos del lector, pero suficientemente larga para llenar la página apropiadamente.

Con sistemas WYSIWYG, los autores a menudo generan documentos agradables estéticamente pero con muy poca o muy inconsistente estructura.  $\text{\LaTeX}$  impide tales errores de formato forzando al autor a declarar la estructura *lógica* del documento.  $\text{\LaTeX}$  escoge entonces la composición más adecuada.

### 1.2.3. Ventajas y desventajas

Cuando gente del mundo WYSIWYG se encuentra con usuarios de  $\text{\LaTeX}$ , a menudo discuten “las ventajas de  $\text{\LaTeX}$  sobre un procesador de textos normal” o lo contrario. Lo mejor que puede hacer cuando un debate tal comienza es mantenerse al margen, pues tales discusiones a menudo se salen de control. Pero a veces uno no puede escapar.

Pues he aquí algunos argumentos. Las principales ventajas de  $\text{\LaTeX}$  sobre procesadores de texto normales son las siguientes:

- Se dispone de composiciones diseñadas profesionalmente, lo que hace que un documento parezca realmente “impreso”.
- El soporte para la composición para la composición de fórmulas matemáticas es muy adecuado.

---

<sup>1</sup>What you see is what you get: Lo que ve es lo que consigue

- Los usuarios sólo tienen que aprender unas pocas órdenes fáciles de entender, que especifican la estructura lógica del documento. Casi nunca necesitan preocuparse del aspecto real del documento.
- Es fácil generar incluso estructuras complejas, como notas al pie, referencias, índices o bibliografías.
- Existen paquetes libres (incluso gratuitos) que facilitan muchas tareas tipográficas especializadas, no soportadas directamente por el  $\text{\LaTeX}$  básico. Por ejemplo, hay disponibles paquetes para incluir gráficos o para componer bibliografías según normas precisas.
- $\text{\LaTeX}$  incita a los autores a escribir textos bien estructurados, porque así trabaja  $\text{\LaTeX}$ .
- $\text{\LaTeX}$  es libre y muy portable. Por tanto, puede ejecutarse en casi cualquier plataforma informática disponible.

$\text{\LaTeX}$  tiene también algunas desventajas, y supongo que me es un poco encontrar alguna notable, aunque estoy seguro de que otros le podrán hablar de cientos

- Aunque pueden ajustarse algunos parámetros dentro de una cierta composición del documento, el diseño de una nueva composición completa es difícil y lleva mucho tiempo.
- Puede que su aprendiz nunca llegue a entender, a pesar de ciertos primeros pasos prometedores, a comprender el concepto de marcado lógico.

### 1.3. Estructura del archivo de entrada

La entrada para  $\text{\LaTeX}$  es un archivo de texto puro. Puede crearlo con cualquier editor de texto. Contiene el texto del documento, así como las órdenes que dirán a  $\text{\LaTeX}$  cómo componer el texto.

Cuando  $\text{\LaTeX}$  procesa un archivo de entrada, espera que siga una cierta estructura. Así, todo archivo de entrada ha de comenzar con la orden

```
\documentclass{...}
```

Esto indica qué tipo de documento pretende escribir. Después, puede incluir órdenes que influyen el estilo de todo el documento, o puede cargar paquetes que añaden nuevas prestaciones al sistema  $\text{\LaTeX}$ . Para cargar un paquete use la orden

```
\usepackage{...}
```

Cuando todo el trabajo de preparación está hecho, comience a escribir el cuerpo del texto con la orden

```
\begin{document}
```

El área entre `\documentclass` y `\begin{document}` se llama *preámbulo*.

Ahora escriba el texto mezclado con órdenes  $\text{\LaTeX}$  útiles. Al final del documento añada la orden

```
\end{document}
```

que dice a  $\text{\LaTeX}$  que termine el trabajo. Cualquier cosa que siga a esta orden será ignorada.

La figura 1.1 muestra el contenido de un archivo  $\text{\LaTeX}$ .

---

```
\documentclass{article}
\begin{document}
Mi primer documento en \LaTeX.
\end{document}
```

---

Figura 1.1: Un archivo  $\text{\LaTeX}$  mínimo.

### 1.3.1. Una típica sesión de consola

Como se decía antes,  $\text{\LaTeX}$  por sí mismo viene sin GUI (interfaz gráfica de usuario) ni botones para pulsar. Es un programa de procesamiento que “mastica” y “digiere” su archivo de entrada para “generar” su(s) archivo(s) de salida. Algunas instalaciones de  $\text{\LaTeX}$  ofrecen una interfaz gráfica donde usted puede escribir y compilar su archivo de entrada (TeXnicCenter, TexMaker, Kile). En otros sistemas puede requerirse la escritura de ciertas órdenes, de modo que he aquí cómo lograr que  $\text{\LaTeX}$  compile su archivo de entrada en un sistema basado en texto.

1. Edite/cree su archivo de entrada  $\text{\LaTeX}$ . Este archivo debe ser texto puro. Puede crearlo con cualquier editor de texto, pero asegúrese de que guarda el archivo con formato *Texto Puro*. Al escoger un nombre para el archivo, póngale como extensión `.tex`
2. Ejecute  $\text{\LaTeX}$  en su archivo de entrada. Si tiene éxito aparecerá un archivo `.dvi`. Puede que necesite ejecutar  $\text{\LaTeX}$  varias veces para que los índices y todas las referencias internas queden correctamente definidas. Si su archivo de entrada tiene un error,  $\text{\LaTeX}$  se lo dirá y parará el procesamiento de su archivo de entrada. Escriba `CTRL + D` para volver a la línea de órdenes.

```
latex miDocumento.tex
```

3. Finalmente, la orden  $\text{\PDF\LaTeX}$  le permite compilar el archivo directamente en pdf.

```
pdflatex miDocumento
```

### 1.3.2. Espacio

- $\text{\LaTeX}$  trata los caracteres “en blanco”, tales como el espacio en blanco o el tabulador, uniformemente como “espacio”.
- Varios caracteres consecutivos en blanco se tratan como uno solo.
- Una línea vacía entre dos líneas de texto define el fin de un párrafo.
- Varias líneas vacías se tratan igual que *una sola* línea vacía.

A la izquierda está el texto del archivo de entrada, y a la derecha está la salida formateada.

#### Ejemplo 1.1.

No importa si usted deja  
uno o varios               espacios  
tras una palabra.

Una línea vacía comienza  
un nuevo párrafo.  
.....

*No importa si usted deja uno o varios  
espacios tras una palabra.  
Una línea vacía comienza un nuevo párrafo.*

### 1.3.3. Caracteres especiales

Los siguientes símbolos son caracteres reservados que tienen un significado especial bajo  $\text{\LaTeX}$ . Si los pone directamente en su texto, normalmente no se imprimirán, sino que obligarán a  $\text{\LaTeX}$  a hacer cosas que no se pretendían.

#### Observación 1 (Caracteres reservados).

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

Como verá, se pueden usar estos caracteres en sus documentos añadiendo una diagonal invertida como prefijo:

#### Ejemplo 1.2.

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

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

Los demás símbolos y muchos más pueden imprimirse con órdenes especiales en fórmulas matemáticas, o como acentos.

**Observación 2** (Contradiagonal).

La contra diagonal (diagonal invertida) no puede introducirse añadiendo otra contradiagonal delante (`\`); esta orden se usa para saltar de línea<sup>a</sup>.

<sup>a</sup>Se debe usar la orden `\backslash` en su lugar.

**1.3.4. Comandos en  $\text{\LaTeX}$** 

Las órdenes  $\text{\LaTeX}$  son sensibles a mayúsculas, y adoptan uno de los dos formatos siguientes:

- Comienzan con una contradiagonal y luego tienen un nombre que consiste sólo en letras. Los nombres de orden terminan con un espacio, un número o cualquier otra “no-letra”.
- Consisten en una contradiagonal y exactamente una no-letra.

**Observación 3.**

$\text{\LaTeX}$  prescinde del espacio en blanco tras las órdenes. Si quiere conseguir un espacio tras una orden, tiene que poner `{ }`.

**Ejemplo 1.3.**

He leído que Knuth divide a la  
gente que trabaja con `\TeX{}` en  
`\TeX{}`nicos y `\TeX` pertos.`\`  
Hoy es `\today`.

*He leído que Knuth divide a la gen-  
te que trabaja con  $\text{\TeX}$  en  $\text{\TeX}$ nicos y  
 $\text{\TeX}$ pertos.  
Hoy es 30 de junio de 2015.*

Algunas órdenes requieren un argumento, que tiene que ponerse entre llaves `{ }` tras el nombre de la orden. Algunas órdenes soportan argumentos opcionales, que se añaden tras el nombre de la orden entre corchetes `[ ]`.

**Ejemplo 1.4.** *Los siguientes ejemplos usan algunas órdenes  $\text{\LaTeX}$ , no se preocupe por ellos; se explicarán más adelante:*

Puede `\textit{fiarse}` de mí.

Por favor, comienza una nueva  
línea justo aquí`\newline` Gracias

*Puede fiarse de mí.  
Por favor, comienza una nueva línea  
justo aquí  
Gracias*

### 1.3.5. Comentarios

Cuando L<sup>A</sup>T<sub>E</sub>X encuentra un caracter % al procesar un archivo de entrada, prescinde del resto de la línea actual, el salto de línea y todo el espacio en blanco al comienzo de la línea siguiente.

Esto puede usarse para escribir notas en el archivo de entrada, que no se mostrán en la versión impresa.

#### Ejemplo 1.5.

```
Este es un % estúpido
% Mejor: instructivo <----
ejemplo: Supercal%
ifragilíst%
icoespialidoso
.....
```

*Este es un ejemplo: Supercalifragilísti-  
coespialidoso*

El caracter % también puede usarse para dividir líneas largas en la entrada donde no se permiten espacios ni saltos de línea.

Para comentarios más largos puede usar el entorno `comment` proporcionado por los paquetes `comment` o `verbatim`. Esto significa que tiene que añadir la línea `\usepackage{verbatim}`, o la línea `\usepackage{comment}` al preámbulo del documento, como se explica en el ejemplo siguiente:

#### Ejemplo 1.6.

```
Esto es otro
\begin{comment}
bastante estúpido,
pero útil
\end{comment}
ejemplo para insertar
comentarios en su texto.
```

*Esto es otro ejemplo para insertar co-  
mentarios en su texto.*

#### Observación 4.

Tenga en cuenta que eso no funciona dentro de entornos complejos, como por ejemplo los matemáticos.

## 1.4. El aspecto del documento

### 1.4.1. Clase de documento

La primera información que L<sup>A</sup>T<sub>E</sub>X necesita saber cuando procesa un archivo de entrada es el tipo de documento que el autor quiere crear. Esto se indica con la orden `\documentclass`

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

Aquí *clase* indica el tipo de documento por crear. La tabla 1.1 lista las clases de documentos explicadas en esta introducción. El parámetro *opciones* personaliza el comportamiento de la clase. Las opciones tienen que separarse por comas. Las opciones más comunes para las clases de documento habituales se listan en la tabla 1.2

article	para artículos en revistas científicas, informes breves, documentación de programas, invitaciones, etc.
proc	para actas, basado en la clase article.
minimal	es lo más pequeña posible. Solamente establece un tamaño de página y una fuente (tipo de letra). Se usa principalmente para depurar errores.
report	para informes más largos que contienen varios capítulos, pequeños libros, tesis, etc.
book	para libros reales.
beamer	para diapositivas. La clase usa letras grandes.

Cuadro 1.1: Clases de documento.

**Ejemplo 1.7.** *Un archivo de entrada para un documento L<sup>A</sup>T<sub>E</sub>X podría empezar con la línea*

```
\documentclass[11pt, twoside, a4paper]{article}
```

*que manda a L<sup>A</sup>T<sub>E</sub>X componer el documento como un artículo con un tamaño de fuente básica de once puntos, y producir un documento adecuado para imprimir a doble cara en papel A4.*

.....

### 1.4.2. Paquetes

Mientras escribe su documento, probablemente halle que hay algunas áreas donde el L<sup>A</sup>T<sub>E</sub>X básico no puede resolver su problema. Si quiere incluir gráficos, texto en color o código fuente de un archivo en su documento, necesita mejorar las capacidades de L<sup>A</sup>T<sub>E</sub>X. Tales mejoras se introducen con *paquetes*. Los paquetes se activan con la orden

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

donde *paquete* es el nombre del paquete y *opciones* es una lista de palabras clave que activan funciones especiales del paquete.

### 1.4.3. Proyectos grandes

Cuando trabaje en proyectos grandes, puede servirle dividir el archivo de entrada en varias partes que puede reunir al compilarlo. La forma más práctica de hacerlo es:

```
\input{nombre-de-archivo}
```



---

<code>10pt, 11pt, 12pt</code>	Establece el tamaño de la principal fuente del documento. Si no se especifica ninguna opción, se aplica 10pt.
<code>a4paper, letterpaper</code>	Define el tamaño del papel. el tamaño por default es letterpaper.
<code>fleqn</code>	Dispone las fórmulas destacadas hacia la izquierda en vez de centradas.
<code>leqno</code>	Coloca los números de las fórmulas a la izquierda en vez de a la derecha.
<code>titlepage, notitlepage</code>	Indica si tras el título del documento debe empezarse una página nueva o no. La clase <b>article</b> no comienza página nueva por omisión, mientras que <b>report</b> y <b>book</b> sí la tienen.
<code>onecolumn, twocolumn</code>	Dice a L <sup>A</sup> T <sub>E</sub> X que componga el documento de una, o dos, columnas respectivamente.
<code>openright</code>	Hace que los capítulos comiencen en páginas derechas. Esto no funciona con la clase <b>article</b> , pues no entiende de capítulos.

---

Cuadro 1.2: Opciones de clases de documento.

---

## Capítulo 2

# Composición de texto

### 2.1. La estructura del texto y el idioma

El objetivo de escribir un texto (salvo cierta literatura moderna) es comunicar ideas, información o conocimiento al lector. El lector entenderá mejor el texto si dichas ideas están bien estructuradas, y verá y sentirá dicha estructura mucho mejor si la forma tipográfica refleja la estructura lógica y semántica del contenido.

L<sup>A</sup>T<sub>E</sub>X se diferencia de otros sistemas de composición en que sólo tiene que decirle tal estructura. La forma tipográfica del texto se deriva según las “reglas” dadas en el archivo de clase del documento y en los varios archivos de estilo usados.

### 2.2. Saltos de línea y de página

Los libros se suelen componer con líneas de igual longitud. L<sup>A</sup>T<sub>E</sub>X inserta los saltos de línea y los espacios necesarios entre palabras optimizando el contenido de todo un párrafo. Si es necesario, también divide palabras con guiones si no caben bien en una línea. Cómo se componen los párrafos dependen de la clase del documento. Normalmente la primer línea de un párrafo lleva sangría, y no hay espacio adicional entre dos párrafos.

En casos concretos puede ser necesario ordenar a L<sup>A</sup>T<sub>E</sub>X que salte de línea:

`\\ ó \newline`

Para comenzar una nueva página usamos:

`\newpage`

## 2.3. Símbolos y caracteres especiales

### 2.3.1. Comillas

#### Observación 1.

No use los símbolos: “ ” para las comillas como haría con una máquina de escribir.

En tipografía hay comillas especiales de apertura y cierre. En  $\text{\LaTeX}$ , use dos acentos graves para abrir comillas y dos apóstrofos para cerrar.

#### Ejemplo 2.1.

‘‘Por favor, pulse la tecla ‘x’.’’

*“Por favor, pulse la tecla ‘x’.”*

### 2.3.2. Guiones y rayas

$\text{\LaTeX}$  conoce cuatro tipos de guión, uno de los cuales es el signo matemático “menos”. Observe cómo obtenerlos:

#### Ejemplo 2.2.

asutro-húngaro, P-valor\\  
páginas 13--67\\  
sí ---dijo él--- \\  
\$0\$, \$1\$ y \$-1\$

*asutro-húngaro, P-valor*  
*páginas 13–67*  
*sí —dijo él—*  
*0, 1 y −1*

### 2.3.3. Tilde (~)

Se trata de un caracter que aparece a menudo en código informático y direcciones de red. Para generarlo en  $\text{\LaTeX}$  se debe usar  $\backslash \sim$  pero el resultado no es realmente lo que se busca:

#### Ejemplo 2.3.

`http://www.rich.edu/\~{}rockefeller`  
`http://www.clever.edu/$\sim$tesla`

*http://www.rich.edu/~rockefeller*  
*http://www.clever.edu/~tesla*

### 2.3.4. Símbolo de grado (°)

El siguiente ejemplo muestra cómo imprimir un símbolo de grado en  $\text{\LaTeX}$ :

**Ejemplo 2.4.**

Estamos a  
 $\$-30^{\circ}\$C.$

*Estamos a  $-30^{\circ}C.$*

.....

**2.3.5. Acentos**

$\LaTeX$  soporta el uso de acentos y caracteres especiales para muchos idiomas. La figura ?? muestra todo tipo de acentos aplicados a la letra o. Por supuesto también funcionan con otras letras.

---

ò	\‘o	ó	\’o	ô	\~o	õ	\~o
ō	\=o	ô	\.o	ö	\"o	ç	\c c
ǒ	\u o	ǒ	\v o	ő	\H o	ø	\c o
ȯ	\d o	ȯ	\b o	ōō	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	ı	\l	Ł	\L
ı	\i	Ј	\j	ı	ı’	ı	ı’

---

Cuadro 2.1: Acentos

**2.4. Soporte para otros idiomas**

Cuando escriba documentos en idiomas distintos del español, hay tres áreas en que  $\LaTeX$  tiene que configurarse adecuadamente:

- Todas las cadenas de texto generadas automáticamente tienen que adaptarse al nuevo idioma. Para muchos idiomas, estos cambios pueden llevarse a cabo mediante el paquete `babel`
- $\LaTeX$  necesita saber las reglas de silabación para el nuevo idioma.
- Reglas tipográficas específicas del idioma. En francés, por ejemplo, hay un espacio obligatorio antes de cada caracter de dos puntos.

Si su sistema ya está configurado adecuadamente, puede activar el paquete `babel` añadiendo la orden

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

tras la orden `\documentclass`. Puede listar los *idiomas* construidos en su sistema  $\text{\LaTeX}$  cada vez que se ejecuta el compilador. Babel activará automáticamente las reglas de silabación para el idioma que escoja.

Para usar el teclado en castellano más común escribimos en el preámbulo:

**Observación 2.**

```
\usepackage[spanish]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

## 2.5. Títulos, capítulos y secciones

Para ayudar al lector a orientarse en su libro, debería dividirlo en capítulos, secciones y subsecciones.  $\text{\LaTeX}$  mediante órdenes especiales que toman el título de la sección como argumento. Es tarea suya al usarlos en el orden correcto.

Las siguientes órdenes de sección están disponibles para la clase **article**:

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

Si quiere dividir su documento en partes sin influir en la numeración de secciones o capítulos puede usar

```
\part{...}
```

Cuando trabaje con las clases **report** o **book**, estará disponible una orden de sección adicional

```
\chapter{...}
```

Como la clase **article** no entiende de capítulos, es muy fácil añadir artículos como capítulos a un libro. El espacio entre secciones, la numeración y el tamaño de fuente de los títulos quedará correctamente establecido por  $\text{\LaTeX}$ .

Dos órdenes de sección son algo especiales:

- La orden `\part` no modifica la secuencia de numeración de los capítulos.
- La orden `\appendix` no toma ningún argumento. Solamente cambia la numeración de capítulos de números a letras.

### 2.5.1. Índice

L<sup>A</sup>T<sub>E</sub>X crea un índice general tomando los encabezados de sección y los números de página del último ciclo de compilación del documento. La orden

```
\tableofcontents
```

sitúa el índice general en el lugar en que se ejecuta la orden. Un documento nuevo debe compilarse dos veces para conseguir un índice general correcto.

Todas las órdenes de sección listadas anteriormente tienen una versión “estrella”. Se trata de órdenes con el mismo nombre pero seguido de un asterisco \*. Generan encabezados de sección que no aparecen en el índice general y que no se numera. La orden `\section{Ayuda}`, por ejemplo, tendría una versión estrella así: `\section*{Ayuda}`.

Normalmente los encabezados aparecen en el índice general exactamente como se introducen en el texto. A veces no es posible, porque el encabezado es demasiado largo y no cabe en el índice general. La entrada para el índice general puede indicarse como un argumento opcional antes del encabezado real.

```
\chapter[Título para el índice]{Un largo  
y aburrido título que aparecerá en el texto}
```

### 2.5.2. Portada

El título de todo el documento se genera con la orden:

```
\maketitle
```

El contenido del título tiene que definirse mediante las órdenes:

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

antes de llamar a `\maketitle`. En el argumento de `\author`, puede poner nombres separados por `\and`.

## 2.6. Referencias cruzadas

En libros, informes y artículos, hay a menudo referencias cruzadas a figuras, cuadros y fragmentos especiales de texto. L<sup>A</sup>T<sub>E</sub>X proporciona las siguientes órdenes para referenciar:

```
\label{marcador}, \ref{marcardor}, \pageref{marcador}
```

donde *marcador* es un identificador escogido por el usuario. L<sup>A</sup>T<sub>E</sub>X reemplaza `\ref` por el número de la sección, subsección, figura, tabla o teorema tras el que se sitúa la orden `\label` correspondiente. El comando `\pageref` imprime el número de la página donde la orden `\label` se sitúa. Como para los títulos de sección, se usan los números de la compilación previa.

**Ejemplo 2.5.**

Una referencia a esta subsección

`\label{etiquetaSeccion}` aparece así:  
 ‘‘ver sección `\ref{etiquetaSeccion}` en  
 la página `\pageref{etiquetaSeccion}`.’’

*Una referencia a esta subsección aparece  
 así: “ver sección 2.5 en la página 6.”*

.....

## 2.7. Notas al pie

Con la orden

`\footnote{texto al pie}`

se imprime una nota al pie de la página actual. Deben ponerse las notas tras la palabra u oración a la que se refieren. Las notas que se refieran a una sentencia o parte de ella deben por tanto ponerse tras la coma o el punto.

**Ejemplo 2.6.**

Las notas al pie `\footnote{Esto  
 es una nota al pie.}` se usan  
 mucho en L<sup>A</sup>T<sub>E</sub>X.

*Las notas al pie<sup>a</sup> se usan mucho en  
 L<sup>A</sup>T<sub>E</sub>X.*

<sup>a</sup>Esto es una nota al pie.

.....

## 2.8. Palabras enfatizadas

Si un texto se escribe a máquina las palabras importantes se enfatizan subrayándolas.

`\underline{texto que quiero subrayar}`

En los libros impresos, sin embargo, las palabras se enfatizan componiéndolas con una fuente *cursiva*. L<sup>A</sup>T<sub>E</sub>X proporciona el comando:

`\emph{texto que quiero en cursiva}`

para enfatizar texto.

### Ejemplo 2.7.

```
\emph{Si usa énfasis en un
fragmento de texto ya
enfatzizado, entonces
\LaTeX{}} usa la fuente
\emph{normal} para
enfatzizar.}
```

Si usa énfasis en un fragmento de texto ya enfatzizado, entonces  $\text{\LaTeX}$  usa la fuente *normal* para enfatzizar.

.....

### Ejemplo 2.8. Algunas otras opciones son:

```
\textit{También puede
\emph{enfatzizar} texto
aunque esté en cursiva,}
\textsf{en fuente \emph{sin serifado},}
\texttt{o en estilo
\emph{mecanogr\'afico}.}
\end{ej}
```

*También puede enfatzizar texto aunque esté en cursiva, en fuente sin serifado, o en estilo mecanográfico.*

.....

## 2.9. Entornos

```
\begin{entorno}
texto
\end{entorno}
```

Aquí la palabra *entorno* es un nombre de entorno. Los entornos pueden anidarse uno dentro de otro mientras se mantega el orden correcto:

```
\begin{aaa}
\begin{bbb}

\end{bbb}
\end{aaa}
```

En las siguientes secciones se explican todos los entornos importantes.



### 2.9.1. Listas

El entorno `itemize` es adecuado para listas simples, el entorno `enumerate` para listas enumeradas y el entorno `description` para descripciones.

**Ejemplo 2.9.** *Una lista simple:*

```
\begin{itemize}
\item Aceite
\item Café
\item Detergente
\item Pan
\end{itemize}
```

- *Aceite*
- *Café*
- *Detergente*
- *Pan*

.....

**Ejemplo 2.10.** *Pasos del método científico:*

```
\begin{enumerate}
\item Observaci'on
\item Hip'otesis
\item Experimentaci'on
\item Teor'ia
\item Ley
\end{enumerate}
```

1. *Observación*
2. *Hipótesis*
3. *Experimentación*
4. *Teoría*
5. *Ley*

.....

**Ejemplo 2.11.** *Podemos describir los pasos del método científico:*

```

\begin{description}
\item [Observación] Análisis
sensorial sobre algo que
despierta curiosidad.

\item [Hipótesis] Es la explicación
que se le da al hecho o
fenómeno observado.

\item [Experimentación] Consiste
en probar para verificar la validez
de la hipótesis.

\item [Teoría] Se hacen teorías
de aquellas hipótesis con más
probabilidades de confirmarse.

\item [Ley] Una hipótesis se
convierte en ley cuando
queda demostrada mediante
la experimentación.
\end{description}

```

**Observación** *Análisis sensorial sobre algo que despierta curiosidad.*

**Hipótesis** *Es la explicación que se le da al hecho o fenómeno observado.*

**Experimentación** *Consiste en probar para verificar la validez de la hipótesis.*

**Teoría** *Se hacen teorías de aquellas hipótesis con más probabilidades de confirmarse.*

**Ley** *Una hipótesis se convierte en ley cuando queda demostrada mediante la experimentación.*

### 2.9.2. Alineación

Los entornos `flushleft` y `flushright` generan párrafos alineados a la izquierda o a la derecha respectivamente. El entorno `center` genera texto centrado. Si no indica los saltos de línea mediante `\\`, `LATEX` los determinará automáticamente.

#### Ejemplo 2.12.

```

\begin{flushleft}
Este texto est\'a alineado a
la izquierda. \LaTeX{} no trata
de justificar las líneas, así
que así quedan.
\end{flushleft}

```

*Este texto está alineado a la izquierda. L<sup>A</sup>T<sub>E</sub>X no trata de justificar las líneas, así que así quedan.*

**Ejemplo 2.13.**

```
\begin{flushright}
Texto alineado\\a la derecha.
\LaTeX{} no trata de
justificar las líneas.
\end{flushright}
```

*Texto alineado  
a la derecha.  $\text{\LaTeX}$  no trata de  
justificar las líneas.*

**Ejemplo 2.14.**

```
\begin{center}
En el centro\\de la Tierra
\end{center}
```

*En el centro  
de la Tierra*

**2.9.3. Citas**

El entorno `quote` es útil para citas, frases importantes y ejemplos.

**Ejemplo 2.15.**

Una regla empírica tipográfica para la longitud de renglón es:

```
\begin{quote}
En promedio, ningún renglón
debería tener más de 66 signos.
\end{quote}
```

Por ello las páginas de  $\text{\LaTeX}$  tienen márgenes tan anchos por default, y los periódicos usan múltiples columnas.

*Una regla empírica tipográfica para la longitud de renglón es:*

*En promedio, ningún renglón debería tener más de 66 signos.*

*Por ello las páginas de  $\text{\LaTeX}$  tienen márgenes tan anchos por default, y los periódicos usan múltiples columnas.*

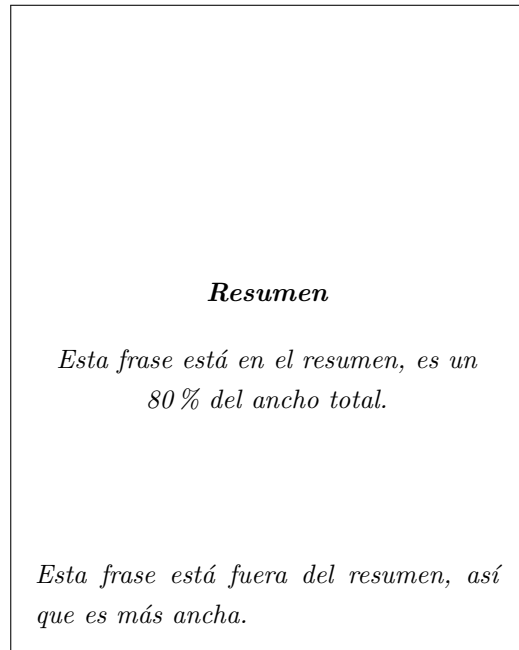
**2.9.4. Resumen (abstract)**

En publicaciones científicas es habitual empezar con un resumen que da al lector una idea rápida de lo que puede esperar.  $\text{\LaTeX}$  proporciona el entorno `abstract` con este propósito. Normalmente `abstract` se usa para documentos compuestos con la clase `article`

### Ejemplo 2.16.

```
\begin{abstract}
Esta frase está en el resumen,
es un 80\% del ancho total.
\end{abstract}

Esta frase está fuera del
resumen, así que es más ancha.
```



### 2.9.5. Tablas

El entorno `tabular` se usa para componer tablas con líneas opcionales horizontales o verticales.  $\text{\LaTeX}$  determina el ancho de las columnas automáticamente.

El argumento *espec* de la orden

```
\begin{tabular}[pos]{espec}
```

define el formato de la tabla. Use un `l` para una columna de texto alineado por la izquierda, `r` para alineación por la derecha y `c` para texto centrado; el comando `p{anchura}` para una columna con texto justificado con saltos de renglón y `|` para una línea vertical.

#### Observación 3.

Si el texto de una columna es demasiado ancha para la página,  $\text{\LaTeX}$  no lo partirá automáticamente. Mediante `p{anchura}` puede definir un tipo de columna especial que partirá el texto como en un párrafo normal.

El argumento *pos* indica la posición vertical de la tabla relativa a la base del texto alrededor. Use una de las lebras `t`, `b` o `c` para indicar la alineación por lo alto, por lo bajo o por el centro,

respectivamente.

Dentro de un entorno `tabular`, el símbolo `&` salta a la columna siguiente, `\\` comienza un nuevo renglón, y `\hline` inserta una línea horizontal. Puede añadir líneas parciales usando `\cline{i-j}` donde  $i$  y  $j$  son los números de las columnas sobre las que debería extenderse la línea.

**Ejemplo 2.17.** Una tabla simple sin líneas:

```
\begin{tabular}{ l c r }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

1	2	3
4	5	6
7	8	9

**Ejemplo 2.18.** Tabla con líneas verticales:

```
\begin{tabular}{ l | c || r }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

1	2	3
4	5	6
7	8	9

**Ejemplo 2.19.** Tabla con líneas horizontales:

```
\begin{tabular}{ l | c || r }
\hline
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\hline
\end{tabular}
```

1	2	3
4	5	6
7	8	9

**Ejemplo 2.20.** Tabla con todas las líneas horizontales:

```
\begin{tabular}{ l | c || r }
\hline
  1 & 2 & 3 \\
\hline
  4 & 5 & 6 \\
\hline
  7 & 8 & 9 \\
\hline
\end{tabular}
```

1	2	3
4	5	6
7	8	9

**Ejemplo 2.21.** *Tabla con líneas horizontales incompletas:*

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\
\cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	<i>hexadecimal</i>
3700	<i>octal</i>
11111000000	<i>binary</i>
1984	<i>decimal</i>

.....

**Ejemplo 2.22.** *Tabla grande sin especificar tamaño:*

```
\begin{tabular}{| l | l | l | l |}
\hline
Day & Min Temp & Max Temp & Summary \\
\hline
Monday & 11C & 22C & A clear day with lots of sunshine.
However, the strong breeze will bring down the temperatures. \\
\hline
Tuesday & 9C & 19C & Cloudy with rain, across many northern regions. Clear spells
across most of Scotland and Northern Ireland,
but rain reaching the far northwest. \\
\hline
Wednesday & 10C & 21C & Rain will still linger for the morning.
Conditions will improve by early afternoon and continue
throughout the evening. \\
\hline
\end{tabular}
```

<i>Day</i>	<i>Min Temp</i>	<i>Max Temp</i>	<i>Summary</i>
<i>Monday</i>	<i>11C</i>	<i>22C</i>	<i>A clear day with lots of sunshine. However, the strong breeze will bring down the temperatures.</i>
<i>Tuesday</i>	<i>9C</i>	<i>19C</i>	<i>Cloudy with rain, across many northern regions. Clear spells across most of Scotland and Northern Ireland, but rain reaching the far northwest.</i>
<i>Wednesday</i>	<i>10C</i>	<i>21C</i>	<i>Rain will still linger for the morning. Conditions will improve by early afternoon and continue throughout the evening.</i>

**Ejemplo 2.23.** La tabla del ejemplo anterior, pero con ancho de columna específico:

```
\begin{tabular}{| l | l | l | p{5cm} |}
\hline
Day & Min Temp & Max Temp & Summary \\
\hline
Monday & 11C & 22C & A clear day with lots of sunshine.
However, the strong breeze will bring down the temperatures. \\
\hline
Tuesday & 9C & 19C & Cloudy with rain, across many northern regions. Clear
spells across most of Scotland and Northern Ireland,
but rain reaching the far northwest. \\
\hline
Wednesday & 10C & 21C & Rain will still linger for the morning.
Conditions will improve by early afternoon and continue
throughout the evening. \\
\hline
\end{tabular}
```

<i>Day</i>	<i>Min Temp</i>	<i>Max Temp</i>	<i>Summary</i>
<i>Monday</i>	<i>11C</i>	<i>22C</i>	<i>A clear day with lots of sunshine. However, the strong breeze will bring down the temperatures.</i>
<i>Tuesday</i>	<i>9C</i>	<i>19C</i>	<i>Cloudy with rain, across many northern regions. Clear spells across most of Scotland and Northern Ireland, but rain reaching the far northwest.</i>
<i>Wednesday</i>	<i>10C</i>	<i>21C</i>	<i>Rain will still linger for the mor- ning. Conditions will improve by early afternoon and continue th- roughout the evening.</i>

#### Observación 4.

El material compuesto con el entorno `tabular` siempre permanece junto en una misma página. Si quiere componer tablas largas, debe usar el entorno `longtable`.



## 2.10. Objetos flotantes

Actualmente la mayoría de las publicaciones contienen muchas figuras y cuadros. Estos elementos requieren un tratamiento especial, porque no pueden dividirse entre dos páginas. Un método posible sería empezar una nueva página cada vez que una figura o un cuadro es demasiado grande para encajar en la página actual. Este enfoque dejaría páginas parcialmente vacías, lo que da mal aspecto.

La solución a este problema es *dejar flotar* cualquier figura o tabla que no encaje en la página actual hacia una página posterior, y rellenar la página actual con texto del documento. L<sup>A</sup>T<sub>E</sub>X ofrece dos entornos para elementos flotantes: uno para cuadros y otro para figuras. Para aprovecharlos bien es importante entender aproximadamente cómo maneja L<sup>A</sup>T<sub>E</sub>X internamente los flotantes. En caso contrario, pueden volverse una fuente de frustraciones, si L<sup>A</sup>T<sub>E</sub>X nunca los pone donde usted quiere que vayan.

Primero veamos las órdenes que L<sup>A</sup>T<sub>E</sub>X proporciona para los flotantes:

Cualquier cosa que vaya adentro de un entorno `figure` o `table` se tratará como flotante. Ambos entornos admiten un parámetro opcional llamado *colocador*

`\begin{figure}[colocador]` ó `\begin{table}[colocador]`

### 2.10.1. Tablas

Este parámetro se usa para decir a L<sup>A</sup>T<sub>E</sub>X dónde puede flotar el elemento. Se construye un *colocador* mediante una cadena de *permisos de flotamiento*. Véase la tabla 2.2

Por ejemplo, una tabla podría empezar con el renglón siguiente:

`\begin{table}[!hbp]`

El colocador `[hbp]` permite que L<sup>A</sup>T<sub>E</sub>X coloque el cuadro justo aquí (h) o abajo (b) en alguna página o en una página especial con flotantes (p), todo ello incluso si no queda tan bien (!). Si no se indica un colocador, las clases típicas suponen `[tbp]`.

Signo	Permiso para flotar.
h	aquí ( <i>here</i> ) en el mismo lugar del texto donde aparece.
t	arriba ( <i>top</i> ) en la página.
b	abajo ( <i>bottom</i> ) en la página.
p	en una <i>página</i> especial sólo con flotantes.
!	sin considerar la mayoría de los parámetros internos que podría impedir su colocación.

Cuadro 2.2: Permisos de deslizamiento.

Una vez explicado lo difícil, quedan más cosas por mencionar sobre los entornos `table` y `figure`. Con la orden

`\caption{texto del pie}`

puede definir un pie para el objeto flotante. L<sup>A</sup>T<sub>E</sub>X añadirá un número correlativo y la palabra “Figura” o “Cuadro”.

Las dos órdenes

`\listoffigures` y `\listoftables`

funcionan análogamente a la orden `\tableofcontents`, imprimiendo un índice de figuras o cuadros, respectivamente. Tales índices muestran los pies completos, así que si tiende a usar pies largos debe tener una versión más corta del pie para los índices. Se consigue poniendo la versión corta entre corchetes tras la orden `\caption`.

`\caption[Corto]{Laaaaaaaaaaaaaargooooooooo}`

Con `\label` y `\ref`, puede crear una referencia al flotante dentro del texto.

#### Observación 5.

Para cambiar la palabra “Cuadro” por “Tabla”, usamos el comando `\renewcommand\tablename{Tabla}`

**Ejemplo 2.24.** Hacemos una tabla cualquiera, encerrada en un entorno `table`:

```
\begin{table}
\begin{tabular}{| c | c | c |}
\hline
(1,1)&(1,2)&(1,3)\\
\hline
(2,1)&(2,2)&(2,3)\\
\hline
(3,1)&(3,2)&(3,3)\\
\hline
\end{tabular}
\caption{Un pie de tabla.}
\label{tablaNumerada}
\end{table}
```

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

Cuadro 2.3: Un pie de tabla.

.....

### 2.10.2. Figuras

La manera fácil de incorporar gráficos en un documento es generarlos en algún programa especializado y después incluir los gráficos acabados en el documento.

Se proporciona un buen conjunto de órdenes para incluir gráficos en el paquete **graphicx**.

1. Cargamos el paquete `graphicx` en el preámbulo del archivo de entrada con:

```
\usepackage{graphicx}
```

2. Creamos (conseguimos) la figura que queremos insertar.

3. Usamos el comando

```
\includegraphics[clave=valor, clave=valor, ...]{archivo}
```

para incluir el *archivo* en el documento. El parámetro opcional acepta una lista separada por comas de *claves* y *valores* asociados. Las *claves* pueden usarse para alterar la anchura, altura y giro del gráfico incluido. La tabla 2.4 lista las claves más importantes.

<code>width</code>	escala el gráfico a la anchura indicada.
<code>height</code>	escala el gráfico a la altura indicada.
<code>angle</code>	gira el gráfico en sentido antihorario.
<code>scale</code>	escala el gráfico.

Cuadro 2.4: Nombres de clave para el paquete `graphicx`.

**Ejemplo 2.25.** *El siguiente código de ejemplo puede ayudar a aclarar las cosas:*

```
\begin{figure}
\includegraphics[scale=0.5]{imagen01}
\caption{Relación trigonométrica.}
\label{primerImagen}
\end{figure}
```

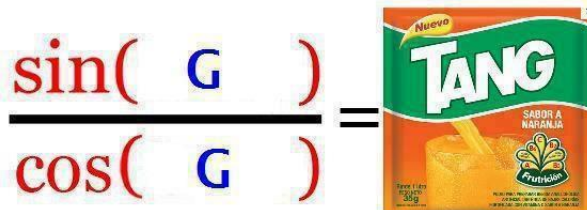


Figura 2.1: Relación trigonométrica.

**Ejemplo 2.26.** *El siguiente código muestra un rotación de la imagen:*

```
\begin{figure}
\includegraphics[angle=90,
                  scale=0.3]{imagen01}
\caption{Relación trigonométrica (Rotación).}
\label{imgRotacion}
\end{figure}
```



Figura 2.2: Relación trigonométrica (Rotación).

**Ejemplo 2.27.** *Hacemos referencia a las imágenes con el comando `\ref{key}`:*

La figura “normal” es la figura `\ref{primerImagen}`, y su rotación se muestra en la figura `\ref{imgRotacion}`.

La figura “normal” es la figura 2.1, y su rotación se muestra en la figura 2.2.

## 2.11. Bibliografía

Se puede crear una bibliografía con el entorno `thebibliography`. Cada entrada empieza con:

```
\bibitem[etiqueta]{marcador}
```

El *marcador* se usa para citar el libro o artículo desde el documento con `\cite{marcador}`

Si no se usa la opción *etiqueta*, las entradas se numerarán automáticamente. El parámetro tras la orden `\begin{thebibliography}` define cuánto espacio reservar para el número de las etiquetas. En el próximo ejemplo, {99} le dice a L<sup>A</sup>T<sub>E</sub>X que espere que ninguno de esos números será más ancho que el número 99.

**Ejemplo 2.28.** *Preferentemente, la bibliografía debe de ir en un archivo aparte:*

```
\begin{thebibliography}{99}
\bibitem{libroEjemplo} Álgebra, Baldor.
\end{thebibliography}
```

*Finalmente, dentro del texto, podemos hacer referencia a algún ítem de la bibliografía con:*

```
\cite{libroEjemplo}.
```

.....

## 2.12. Fuentes y tamaños

L<sup>A</sup>T<sub>E</sub>X escoge la fuente y el tamaño de fuente apropiados basándose en la estructura lógica del documento (secciones, notas al pie, etc.). En algunos casos, quizá desee cambiar fuentes y tamaños a mano. Para hacerlo, puede usar las órdenes listadas en las tablas 2.5 y 2.6. El tamaño real de cada fuente es una cuestión de diseño y depende de la clase de documento y de sus opciones. La tabla 2.7 muestra los tamaños absolutos en puntos para estas órdenes según se implementan en las clases de documentos normales.

**Ejemplo 2.29.**

```
{\small Pequeña \textbf{negrita}
del África tropical,}
{\Large grande y \textit{cursiva}va
eres tú ya.}
```

*Pequeña **negrita** del África tropical,  
grande y cursiva eres tú ya.*

.....

En relación a las órdenes de tamaño de fuente, las llaves representan un papel significativo. Se usan para construir *grupos*. Los grupos limitan el alcance de la mayoría de las órdenes de L<sup>A</sup>T<sub>E</sub>X.

**Ejemplo 2.30.**

```
Adora los {\Large grandes y
{\small pequeños} placeres}.
```

*Adora los grandes y pequeños placeres.*

.....

<code>\textrm{...}</code>	rematada	<code>\textsf{...}</code>	palo seco
<code>\texttt{...}</code>	de máquina	<code>\emph{...}</code>	<i>énfasis</i>
<code>\textmd{...}</code>	peso medio	<code>\textbf{...}</code>	<b>negrita</b>
<code>\textup{...}</code>	recta	<code>\textit{...}</code>	<i>cursiva</i>
<code>\textsl{...}</code>	oblicua	<code>\textsc{...}</code>	VERSALITAS

Cuadro 2.5: Fuentes

<code>\tiny</code>	fente minúscula	<code>\Large</code>	más grande
<code>\scriptsize</code>	fente muy pequeña	<code>\LARGE</code>	muy grande
<code>\footnotesize</code>	bastante pequeña	<code>\huge</code>	enorme
<code>\large</code>	fente grande	<code>\Huge</code>	la más

Cuadro 2.6: Tamaños de fuente.

Tamaño	10pt(default)	opción 11pt	opción 12pt
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Cuadro 2.7: Tamaños absolutos en puntos para las clases normales.

---

## Capítulo 3

# Composición de fórmulas matemáticas

### 3.1. Generalidades

L<sup>A</sup>T<sub>E</sub>X tiene un modo especial para componer matemáticas. Hay dos posibilidades: escribir las matemáticas dentro de un párrafo, es decir, en el mismo renglón que el resto del texto, o partir el párrafo para componer las matemáticas aparte, destacadas. El texto matemático *dentro* del párrafo se introduce entre \$ y \$, o \begin{equation} y \end{equation}.

#### Ejemplo 3.1.

Sume \$a\$ al cuadrado y \$b\$ al cuadrado para obtener \$c\$ al cuadrado. Más formalmente:  
\$c^2=a^2+b^2\$.

*Sume a al cuadrado y b al cuadrado para obtener c al cuadrado. Más formalmente:  $c^2 = a^2 + b^2$ .*

.....

Cuando quiera que sus ecuaciones o fórmulas matemáticas más grandes se sitúen destacadas aparte del resto del párrafo, es preferirle *aislarlas*. Para ello, se puede usar \$\$ y \$\$.

#### Ejemplo 3.2.

Suma \$a\$ al cuadrado y \$b\$ al cuadrado para obtener \$c\$ al cuadrado. Más formalmente:  
\$\$c^2=a^2+b^2\$\$

*Suma a al cuadrado y b al cuadrado para obtener c al cuadrado. Más formalmente:*

$$c^2 = a^2 + b^2$$

Si quiere que L<sup>A</sup>T<sub>E</sub>X enumere sus ecuaciones, puede usar el entorno `equation`. Puede etiquetar mediante `\label` la ecuación con un número y referirse a éste desde otro lugar del texto usando `\ref`:

**Ejemplo 3.3.**

```
\begin{equation}
\epsilon > 0
\label{epsilon}
\end{equation}
```

De (`\ref{epsilon}`), se deduce cualquier cosa.

$$\epsilon > 0 \quad (3.1)$$

*De (??), se deduce cualquier cosa.*

.....

Observe las diferencias de estilo entre las ecuaciones en párrafo y las aisladas:

**Ejemplo 3.4.**

Dentro del texto tenemos:

```
\lim_{n \to \infty} \sum_{k=1}^n
\frac{1}{k^2} = \frac{\pi^2}{6}
```

La misma ecuación se ve:

```
$$\lim_{n \to \infty} \sum_{k=1}^n
\frac{1}{k^2} = \frac{\pi^2}{6}$$
```

*Dentro del texto tenemos:*

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

*La misma ecuación se ve:*

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

.....

Hay diferencias entre modo matemático y modo *texto*. Por ejemplo, en modo matemáticas:

1. La mayoría de los espacios y saltos de línea no significan nada, pues todos los espacios se deducen lógicamente de las expresiones matemáticas, o tienen que ser indicados con órdenes especiales como `\,`, `\quad` o `\qquad`.
2. No se permiten renglones vacíos. Sólo un párrafo por fórmula.
3. Cada letra se considera como nombre de una variable y como tal será escrita así. Si se necesita escribir texto normal dentro de una fórmula (tipo redondo y espaciado normal) entonces tiene que introducir el texto usando las órdenes `\text{rm}{...}`

**Ejemplo 3.5.**

```
\begin{equation}
\forall x \in \mathbf{R}:
\quad x^2 \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R} : \quad x^2 \geq 0 \quad (3.2)$$



```
\begin{equation}
x^2 \geq 0\quad
\text{para todo }
x \in \mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \quad \text{para todo } x \in \mathbf{R} \quad (3.3)$$

## 3.2. Agrupación en modo matemático

La mayoría de las órdenes en modo matemático actúan sólo sobre el siguiente caracter, así que si quiere que una orden afecte a varios caracteres, debe agruparlos juntos entre llaves:  $\{\dots\}$ .

**Ejemplo 3.6.**

```
\begin{equation}
a^{x+y} \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \quad (3.4)$$

## 3.3. Construcción de bloques de una fórmula matemática

### 3.3.1. Alfabeto griego

**Ejemplo 3.7.** *Las letras griegas minúsculas se introducen con su nombre usando la inicial en minúscula:*

```
\lambda, \xi, \pi, \phi, \omega
```

 $\lambda, \xi, \pi, \phi, \omega$ 

*Mientras que las mayúsculas sólo cambian su letra inicial:*

```
\Lambda, \Xi, \Pi, \Phi, \Omega
```

 $\Lambda, \Xi, \Pi, \Phi, \Omega$ 

#### Observación 1.

No todas las letras griegas mayúsculas están definidas. Por ejemplo, alfa ( $\alpha$ ) y mu ( $\mu$ ) mayúsculas.

### 3.3.2. Exponentes y subíndices

**Ejemplo 3.8.** *Los exponentes y subíndices pueden indicarse con los caracteres  $\wedge$  y  $_$ .*

`$a_1$, $x^2$, $e^{-\alpha t}$`

`$a^{12}_{34}$, $e^{x^2} \neq e^{x^2}$`

$$a_1, x^2, e^{-\alpha t}$$

$$a_{34}^{12}, e^{x^2} \neq e^{x^2}$$

.....

**Ejemplo 3.9.** La *raíz cuadrada* se introduce como `\sqrt{...}`; la raíz *n*ésima se genera con `\sqrt[n]{...}`. El tamaño del signo de la raíz lo determina automáticamente *L<sup>A</sup>T<sub>E</sub>X*.

`$$\sqrt{x}$, $\sqrt{x^2+\sqrt{y}}$`

`$$\sqrt[3]{2}$, $\sqrt[4]{x^2+y^2}$`

$$\sqrt{x}, \sqrt{x^2 + \sqrt{y}}$$

$$\sqrt[3]{2}, \sqrt[4]{x^2 + y^2}$$

.....

### 3.3.3. Agrupación

**Ejemplo 3.10.** Las órdenes `\overline{...}` y `\underline{...}` crean *líneas horizontales* justo encima o debajo de una expresión.

`$$\overline{m+n}$`

`$$\underline{a+b+c}$`

$$\overline{m+n}$$

$$\underline{a+b+c}$$

.....

**Ejemplo 3.11.** Los comandos `\overbrace{...}` y `\underbrace{...}` crean *llaves horizontales* encima o bajo una expresión.

`$$\underbrace{a+b+\cdots+z}_{26}$`

`$$\overbrace{1+2+\cdots+9}^9$`

$$\underbrace{a+b+\cdots+z}_{26}$$

$$\overbrace{1+2+\cdots+9}^9$$

### 3.3.4. Operaciones implícitas

**Ejemplo 3.12.** Para añadir acentos matemáticos como flechas o tildes podemos usar el símbolo `'`.

`$$`

`y=x^2 \quad y'=2x \quad y''=2`

`$$`

$$y = x^2 \quad y' = 2x \quad y'' = 2$$

.....

**Ejemplo 3.13.** Los *vectores* suelen indicarse añadiendo flechas pequeñas encima de una variable. Los comandos son: `\overrightarrow{...}` y `\overleftarrow{...}`.

`$$\overrightarrow{AB}$$`

`$$\overleftarrow{BA}$$`

$$\overrightarrow{AB}$$

$$\overleftarrow{BA}$$

**Ejemplo 3.14.** La multiplicación no suele tener un punto explícito; sin embargo, a veces sí se escribe para ayudar a los ojos del lector a agrupar los elementos de una fórmula. Se puede usar `\cdot` o `\times` en estos casos:

`$$v=\sigma_1 \cdot \sigma_2`  
`\tau_1 \times \tau_2$$`

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \times \tau_2$$

### 3.3.5. Funciones trascendentes

Los nombres de funciones como logaritmo suelen componerse en una fuente derecha, y no en cursiva como se hace con las variables, así que L<sup>A</sup>T<sub>E</sub>X proporciona las siguientes órdenes para componer las nombres de funciones más importantes, tanto para documentos en inglés:

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>
<code>\sinh</code>	<code>\sup</code>	<code>\tan</code>	<code>\tanh</code>	<code>\min</code>	<code>\Pr</code>
<code>\sec</code>	<code>\sin</code>				

**Ejemplo 3.15.**

`$$\lim_{x \rightarrow 0} \frac{\sin x}{x}=1$$`

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

### 3.3.6. Símbolos grandes

**Ejemplo 3.16.** El operador *integral* se genera con `\int`, la *sumatoria* con `\sum` y el *producto* con `\prod`. Los límites superior e inferior se indican con `^` y `_` como superíndice y subíndices:

$\sum_{i=1}^n$   
 $\int_a^b$   
 $\prod_{\epsilon}$

$$\sum_{i=1}^n$$

$$\int_a^b$$

$$\prod_{\epsilon}$$

**Ejemplo 3.17.** Si pone la orden `\left` ante un delimitador de apertura, y `\right` ante un delimitador de cierre,  $\text{\LaTeX}$  determinará automáticamente el tamaño correcto del delimitador `\right`, y que el tamaño se determina correctamente sólo si ambos se componen en la misma línea. Si no quiere que aparezca nada a la derecha, use `\righth`.

$1 + \left(\frac{1}{1-x^2}\right)^3$

$$1 + \left(\frac{1}{1-x^2}\right)^3$$

**Ejemplo 3.18.** Hay varias órdenes para introducir **tres puntos** en una fórmula. `\ldots` compone los puntos en la línea de base, `\cdots` los coloca centrados. Además, están las órdenes `\vdots` para puntos verticales y `\ddots` para puntos diagonales.

$x_1, \ldots, x_n$   
 $a_1, \cdots, a_n$

$$x_1, \ldots, x_n$$

$$a_1, \cdots, a_n$$

### 3.4. Material alineado verticalmente

**Ejemplo 3.19.** Para crear **matrices**, use el entorno `array`. Funciona de forma similar al entorno `tabular`.

```


$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$


```

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

**Ejemplo 3.20.** El entorno `array` también puede usarse para componer expresiones que tienen un delimitador grande usando “.” como un delimitador derecho de (`\right`):

```


$$y = \begin{cases} a & \text{si } d > c \\ b+x & \text{por la mañana} \\ l & \text{el resto del día} \end{cases}$$


```

$$y = \begin{cases} a & \text{si } d > c \\ b+x & \text{por la mañana} \\ l & \text{el resto del día} \end{cases}$$

**Ejemplo 3.21.** Al igual que con el entorno `tabular`, puede también dibujar líneas en el entorno `array` para separar los elementos:

```


$$\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}$$


```

$$\left( \begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array} \right)$$

**Ejemplo 3.22.** Para fórmulas que ocupan varios renglones o para sistemas, puede usar los entornos `eqnarray` y `eqnarray*` en lugar de `$$`. En `eqnarray` cada renglón lleva un número de ecuación; en `eqnarray*` no se numera ninguno.

```
\begin{eqnarray}
f(x)&=&\cos x\\
f'(x)&=&-\sin x\\
\int_0^{\pi} f(y)dy&=&\sin x
\end{eqnarray}
```

$$f(x) = \cos x \quad (3.5)$$

$$f'(x) = -\sin x \quad (3.6)$$

$$\int_0^{\pi} f(y)dy = \sin x \quad (3.7)$$

**Observación 2.**

Los entornos `eqnarray` y `eqnarray*` funcionan como una tabla de tres columnas de la forma `{rcl}`, donde la columna del medio puede usarse para el signo *igual*, el signo *distinto* o cualquier otro signo que quiera poner.

**Ejemplo 3.23.** Las *ecuaciones largas* no se dividen automáticamente en trozos adecuados. El autor ha de indicar dónde partirlas y cuánto sangrar los trozos. Los siguientes dos métodos son los más habituales para conseguirlo.

```
\begin{eqnarray}
\sin x&=&x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
&&\frac{x^7}{7!} + \cdots
\end{eqnarray}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \quad (3.8)$$

```
\begin{eqnarray}
\cos x&=&1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \quad (3.9)$$

**Observación 3.**

La orden `\nonumber` dice a  $\text{\LaTeX}$  que no genere un número para la correspondiente ecuación.

### 3.5. Entornos personalizados

Al escribir documentos matemáticos, probablemente necesite una manera de crear “lemas”, “definiciones”, “axiomas” y estructuras similares. Esto se hace con la orden `newtheorem`.

```
\newtheorem{nombre}[contador]{texto}[sección]
```

El argumento *nombre* es una palabra corta usada para identificar el tipo de “teorema”. Con el argumento *texto* se define el nombre real de “teorema”, que aparecerá en el documento final.

Los argumentos entre corchetes son opcionales. Se usan ambos para indicar la numeración usada en el “teorema”. Use el argumento *contador* para indicar el *nombre* de un “teorema” declarado con anterioridad. El nuevo “teorema” tomará sus números.

Tras ejecutar la orden `\newtheorem` en el preámbulo de su documento, puede usar la siguiente orden dentro del documento

```
\begin{nombre}[texto]
Este es mi interesante teorema
\end{nombre}
```

#### Observación 4.

Seguramente  $\text{\LaTeX}$  necesitará el paquete `amsthm` para poder definir nuevos teoremas.

El paquete `amsthm` proporciona la orden `\newtheoremstyle{estilo}` que le permite definir el estilo:

**definition** título en negrita, cuerpo en recta.

**plain** título en negrita, cuerpo en cursiva.

**remark** título en cursiva, cuerpo en recta.

**Ejemplo 3.24.** *Primero definimos, en el preámbulo del documento, las siguientes instrucciones:*

<code>\theoremstyle{definition}</code>	<code>\newtheorem{ley}{Ley}</code>
<code>\theoremstyle{plain}</code>	<code>\newtheorem{jurado}[ley]{Jurado}</code>
<code>\theoremstyle{remark}</code>	<code>\newtheorem*{marg}{Margarita}</code>

```
\begin{ley} \label{ley:caja}
No esconder en la caja negra
\end{ley}
```

```
\begin{jurado}[Los Doce]
¡Podría ser usted! Cuidado y
vea la ley \ref{ley:caja}
\end{jurado}
```

```
\begin{marg}
No, no, no
\end{marg}
```

**Ley 1** *No esconder en la caja negra*

**Jurado 2 (Los Doce)** *¡Podría ser usted! Cuidado y vea la ley ??*

**Margarita** *No, no, no*

#### Observación 5.

El teorema “Jurado” usa el mismo contador que el teorema “Ley”, así que le corresponde un número en secuencia con las otras “Leyes”. El argumento entre corchetes se usa para indicar un título o algo similar para el teorema.