

# **Manual de LaTeX**

Alfredo Sánchez Alberca

6/1/2022

# Índice de contenidos

<b>Prefacio</b>	<b>3</b>
Licencia . . . . .	3
<b>1 Introducción</b>	<b>4</b>
1.1 ¿Qué es $T_{E}X$ ? . . . . .	4
1.2 ¿Qué es $L_{A}T_{E}X$ . . . . .	4
1.3 Instalación . . . . .	5
1.4 Hola $L_{A}T_{E}X$ . . . . .	5
1.4.1 Compilación . . . . .	6
<b>2 Estructura de un documento</b>	<b>9</b>
2.1 Esqueleto básico para <code>pdf<math>\text{latex}</math></code> . . . . .	9
2.1.1 Clase de un documento . . . . .	10
2.1.2 Preámbulo . . . . .	11
2.1.3 Cuerpo . . . . .	11
2.2 Esqueleto básico para <code>x<math>\text{e}</math>latex</code> . . . . .	11
<b>3 Secciones y párrafos</b>	<b>13</b>
3.1 Secciones y subsecciones . . . . .	13
3.2 Párrafos y cambios de línea . . . . .	14
3.3 Justificación . . . . .	15
<b>4 Formateo básico</b>	<b>17</b>
4.1 Negrita, cursiva y subrayado . . . . .	17
4.2 Familias de tipos de letra . . . . .	18
4.3 Perfiles de letra . . . . .	19
4.4 Tamaños de letra . . . . .	20
<b>5 Listas</b>	<b>22</b>
<b>6 Tablas</b>	<b>26</b>
<b>7 Imágenes</b>	<b>29</b>

<b>8</b>	<b>Fórmulas matemáticas</b>	<b>31</b>
8.1	Símbolos matemáticos . . . . .	32
8.1.1	Letras griegas . . . . .	32
8.1.2	Operadores aritméticos . . . . .	33
8.1.3	Relaciones . . . . .	33
8.1.4	Operadores binarios . . . . .	33
8.1.5	Lógica . . . . .	33
8.1.6	Conjuntos . . . . .	33
8.1.7	Flechas . . . . .	33
8.1.8	Otros símbolos . . . . .	34
8.1.9	Funciones . . . . .	34
8.2	Subíndices y superíndices . . . . .	34
8.3	Fracciones . . . . .	36
8.4	Sumatorios, productorios e integrales . . . . .	36
8.5	Sombreros . . . . .	38
8.6	Matrices . . . . .	39
8.7	Teoremas . . . . .	41
<b>9</b>	<b>Entornos flotantes</b>	<b>43</b>
9.1	Entorno flotante para figuras . . . . .	43
9.2	Entorno flotante para tablas . . . . .	44
<b>10</b>	<b>Referencias cruzadas y notas a pie</b>	<b>47</b>
10.1	Referencias cruzadas . . . . .	47
10.2	Notas a pie de página . . . . .	48
	<b>Bibliografía</b>	<b>49</b>

# Prefacio

¡Bienvenida/os al manual de L<sup>A</sup>T<sub>E</sub>X.!

Este libro presenta una introducción al lenguaje de composición de textos L<sup>A</sup>T<sub>E</sub>X con un enfoque orientado a la creación de documentos científicos y técnicos con contenido matemático.

## Licencia

Esta obra está bajo una licencia Reconocimiento – No comercial – Compartir bajo la misma licencia 3.0 España de Creative Commons. Para ver una copia de esta licencia, visite <https://creativecommons.org/licenses/by-nc-sa/3.0/es/>.

Con esta licencia eres libre de:

- Copiar, distribuir y mostrar este trabajo.
- Realizar modificaciones de este trabajo.

Bajo las siguientes condiciones:

- **Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
- **No comercial.** No puede utilizar esta obra para fines comerciales.
- **Compartir bajo la misma licencia.** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Estas condiciones pueden no aplicarse si se obtiene el permiso del titular de los derechos de autor.

Nada en esta licencia menoscaba o restringe los derechos morales del autor.

# 1 Introducción

## 1.1 ¿Qué es $T_{\text{E}}X$ ?

$T_{\text{E}}X$  es un sistema de composición de documentos de alta calidad, orientado especialmente a la creación de documentos científicos y técnicos que incluyen fórmulas matemáticas. Fue creado por [Donald Knuth](#) en 1978.

A diferencia de un procesador de textos como por ejemplo Microsoft Word,  $T_{\text{E}}X$  no es una aplicación sino un lenguaje de programación que requiere compilar el código fuente para obtener el documento final. Esto, que a priori podría parecer una desventaja, en realidad es la gran ventaja de  $T_{\text{E}}X$  frente a los procesadores de texto que siguen el paradigma [WYSIWYG](#) (*What You See Is What You Get*), ya que permite separar fácilmente el contenido y la estructura de un documento, de su formato, de manera que el usuario puede centrarse en el contenido y la estructura del documento, y dejar que  $T_{\text{E}}X$  se encargue del formato. De hecho,  $T_{\text{E}}X$  incorpora un potente [lenguaje de marcado](#) para estructurar y formatear el texto de un documento. Por ejemplo, mientras que para poner una palabra en negrita con un procesador de textos como Microsoft Word, bastaría con seleccionar la palabra y hacer clic en el botón de negrita para ver automáticamente la palabra en negrita en la pantalla del ordenador, en  $T_{\text{E}}X$  habría que escribir en el fichero con el código fuente `\bf palabra` y después compilar el código fuente para obtener un documento final con la palabra en negrita (el comando `\bf`, que permite aplicar la negrita, se conoce como *marca* o *tag* en inglés.)

La página principal con información sobre  $T_{\text{E}}X$  es la del [TeX Users Group](#).

## 1.2 ¿Qué es $\text{\LaTeX}$ ?

$\text{\LaTeX}$  es un conjunto de macros para  $T_{\text{E}}X$  debido originalmente a [Leslie Lamport](#) para facilitar el uso de  $T_{\text{E}}X$ .

Tanto  $T_{\text{E}}X$  como  $\text{\LaTeX}$  son programas de código abierto, liberados bajo la [licencia LPPL](#).

Otra de las grandes ventajas de  $\text{\LaTeX}$  es que existen multitud de paquetes de código libre para generar distintos tipos de documentos que pueden descargarse desde el repositorio [CRAN](#).

La página principal sobre  $\text{\LaTeX}$  es [The LaTeX project](#).

## 1.3 Instalación

Existen distintas distribuciones de  $\text{\LaTeX}$  y algunas de ellas son multiplataforma, es decir, están disponibles para diferentes sistemas operativos. Las distribuciones más comunes son:

- [TeXLive](#) para Windows, Mac OSX y Linux.
- [MiKTeX](#) para Windows, Mac OSX y Linux.
- [MacTeX](#) para Mac OSX.

En sus respectivas páginas está explicado el procedimiento de instalación de cada una.

Junto a la distribución de  $\text{\LaTeX}$ es también habitual instalar algún editor de texto para escribir el código fuente. En realidad puede usarse cualquier editor de texto que ya esté instalado en nuestro sistema operativo, pero los existen entornos de edición especializados que facilitan muchas de las tareas del proceso de composición de documentos con  $\text{\LaTeX}$ . Los más comunes son:

- [TexMaker](#). Es un editor de texto libre, multiplataforma, con muchos asistentes disponibles que permite previsualizar en tiempo real el documento final en pdf
- [Textstudio](#). Es otro editor libre y multiplataforma que incorpora aún más asistentes que el anterior.
- [Vim](#) Es un editor de texto simple de propósito general que también es libre y multiplataforma. Incorpora paquetes o plugins específicos para facilitar la creación de documentos con  $\text{\LaTeX}$ . Especialmente indicado para trabajar desde la terminal.
- [Emacs](#). Es otro editor similar a Vim, muy extendido entre los usuarios que prefieren usar la terminal.
- [Visual Studio Code](#). Es un potente entorno de desarrollo multipropósito. Dispone de paquetes para los lenguajes de programación más comunes, entre ellos  $\text{\LaTeX}$ .

Pero también se puede empezar a componer documentos sin necesidad de instalar nada en el ordenador, usando un editor on-line como por ejemplo [Overleaf](#)

## 1.4 Hola LaTeX

A modo de ejemplo, empezaremos por crear un sencillo documento con el texto “Hola  $\text{\LaTeX}$ ”.

Para ello utilizaremos nuestro editor de texto preferido para crear un fichero de texto con el nombre `main.tex` y el siguiente contenido:

```
\documentclass{article}
\usepackage[spanish]{babel}
\begin{document}
Hola \LaTeX
```

```
\end{document}
```

### ! Importante

El nombre del fichero de texto con el código fuente de  $\text{\LaTeX}$  puede ser el que queramos, pero es importante que la extensión sea `.tex`,

Aunque más adelante se verá la estructura general del código fuente de un documento en  $\text{\LaTeX}$ , a continuación se explica brevemente el contenido de este fichero:

1. En la primera línea se especifica el tipo de documento (`article`).
2. En la segunda línea se especifica el idioma del documento (`spanish`).
3. La tercera línea marca el comienzo del documento.
4. La cuarta línea contiene el texto del documento. `\LaTeX` es un comando que produce la salida  $\text{\LaTeX}$ .
5. La quinta línea marca el final del documento.

## 1.4.1 Compilación

Para obtener el documento final hay que compilar el fichero fuente. Existen diferentes formas de hacerlo y en los editores anteriores suele ser tan sencillo como hacer clic en un botón o pulsar una combinación de teclas, pero en última instancia todos ellos hacen una llamada al compilador de  $\text{\LaTeX}$  que es quien se encarga de convertir el código fuente en el documento final.

Cada distribución de  $\text{\LaTeX}$  viene con varios compiladores. Los más habituales son:

- `latex`: Es el compilador más antiguo y genera documentos en [formato dvi](#), que es un formato independiente creado mucho antes que el formato `pdf`.
- `pdflatex`: Es el compilador más usado y genera documentos en formato `pdf`.
- `xelatex`: Es un compilador más moderno que admite caracteres Unicode en el código fuente y el uso de tipografías más modernas.

En una terminal, la compilación de este documento sería tecleando el comando `latex main.tex`, `pdflatex main.tex` o `xelatex main.tex`, dependiendo del compilador que se quiera usar. A continuación se muestra la salida que genera el compilador `pdflatex` al compilar el fichero `main.tex`.

```
> pdflatex main.tex
s is pdfTeX, Version 3.141592653-2.6-1.40.22 (TeX Live 2021) (preloaded format=pdflatex)
restricted \write18 enabled.
entering extended mode
```

```

(./main.tex
LaTeX2e <2021-06-01> patch level 1
L3 programming layer <2021-06-18>
(/usr/local/texlive/2021/texmf-dist/tex/latex/base/article.cls
Document Class: article 2021/02/12 v1.4n Standard LaTeX document class
(/usr/local/texlive/2021/texmf-dist/tex/latex/base/size10.clo))
(/usr/local/texlive/2021/texmf-dist/tex/generic/babel/babel.sty
(/usr/local/texlive/2021/texmf-dist/tex/generic/babel/babel.def
(/usr/local/texlive/2021/texmf-dist/tex/generic/babel/txtbabel.def))
(/usr/local/texlive/2021/texmf-dist/tex/generic/babel-spanish/spanish.ldf))
(/usr/local/texlive/2021/texmf-dist/tex/latex/l3backend/l3backend-pdftex.def)
(./main.aux) [1{/usr/local/texlive/2021/texmf-var/fonts/map/pdftex/updmap/pdftex.map}]
(./main.aux) )</usr/local/texlive/2021/texmf-dist/fonts/type1/public/amsfonts/cm/cmr10.pfb></usr/local/texlive/2021/texmf-dist/fonts/type1/public/amsfonts/cm/cmr7.pfb>
Output written on main.pdf (1 page, 20106 bytes).
Transcript written on main.log.

```

En la Figura 1.1 se puede apreciar el documento final que se obtiene tras compilar el código fuente en Overleaf.



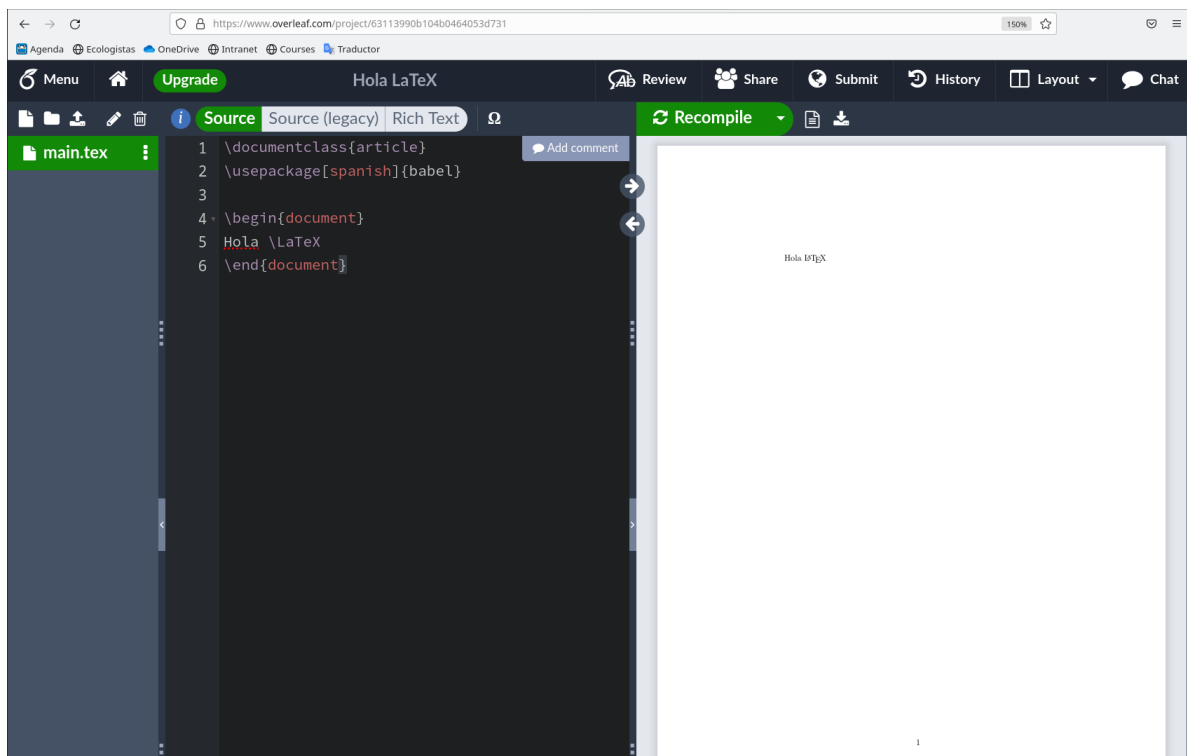


Figura 1.1: Creación de un documento en Overleaf.

## 2 Estructura de un documento

### 2.1 Esqueleto básico para pdf $\text{\LaTeX}$

El esqueleto básico del código fuente de un documento en español para compilar con  $\text{\LaTeX}$  o pdf $\text{\LaTeX}$  es el siguiente:

```
% CLASE
\documentclass[a4paper,10pt]{article}

% PREÁMBULO
% Paquetes
\usepackage[utf8]{inputenc}
\usepackage[spanish]{babel}
\usepackage[T1]{fontenc}

% Título, autor y fecha
\title{Título}
\author{Autor}
\date{Fecha}

% CUERPO
\begin{document}

\maketitle

% Resumen
\begin{abstract}
Resumen
\end{abstract}

% Tabla de contenidos
\tableofcontents

Contenido del documento
```

`\end{document}`

Antes de explicar las distintas partes de este esqueleto conviene mencionar varias cosas sobre la sintaxis de algunos elementos básicos:

- **Comandos:** Los comandos comienzan siempre por la barra invertida (backslash) `\`. En muchas ocasiones van acompañados de argumentos obligatorios que se escriben entre llaves `{...}` y opcionales que se escriben entre corchetes `[...]`.
- **Entornos:** Los entornos, a diferencia de los comandos, son bloques de código sobre los que se aplica alguna acción, y están delimitados siempre por un comando de apertura `\begin{entorno}` y otro de cierre `\end{entorno}`.
- **Comentarios:** Al igual que en otros lenguajes de programación se pueden hacer comentarios en el código fuente que no serán interpretados por el compilador. Para ello se utiliza el símbolo de porcentaje `%` al comienzo del comentario.
- **Símbolos reservados:** Existe una serie de símbolos que están reservados para funciones especiales.
  - `\`: Indica el inicio de un comando.
  - `$`: Declara el entorno matemático.
  - `{ }`: Inicia y finaliza un grupo.
  - `#`: Indica el número de un argumento en la definición de comandos.
  - `%`: Indica el inicio de un comentarios.
  - `&`: Separa elementos en una tabla o fórmula.
  - `^`: Escribe un superíndice.
  - `_`, Escribe un subíndice.
  - `~`, Indica por dónde se puede partir una palabra al final de una línea.

Para que aparezcan estos caracteres en el documento final es necesario escribirlos en el código fuente precedidos por la barra invertida (`\$, \{, \}, \#, \%, \&, \^, \_, \sim`) excepto la barra invertida que se escribe con el comando `\backslash`.

A continuación se explican las partes del esqueleto anterior.

### 2.1.1 Clase de un documento

La primera línea de un fichero con código `LATEX` indica la clase de documento que se va a generar mediante el comando `\documentclass`. En el ejemplo aparece un argumento obligatorio que indica el tipo de documento que se desea crear, artículo (`article`), pero se pueden crear otros tipos de documentos como informes (`report`), libros (`book`) o cartas (`letter`). . Y también aparecen dos argumentos opcionales, `a4paper` que indica el tamaño de la hoja en el documento final (a4), y `10pt` que indica el tamaño base de la fuente utilizada en el documento (existe también `11pt` y `12pt`).

### 2.1.2 Preámbulo

El preámbulo es la parte que va después de la clase y antes del comienzo del cuerpo del documento. En parte suele utilizarse para la carga de los paquetes de macros que se van a utilizar en el documento y la configuración del documento. En el ejemplo el preámbulo comienza con la carga de tres paquetes mediante el comando `\usepackage`: el paquete `inputenc` que permite definir la [codificación de los caracteres](#) del código fuente (conviene utilizar la codificación `utf8` sobre todo si se van a utilizar caracteres no ASCII); el paquete `babel` que permite definir el idioma del documento (`spanish`); y el paquete `fontenc` que especifique las codificaciones<sup>1</sup> de las fuentes (`T1`).

A continuación, se suelen configurar algunos aspectos del documento como podrían ser los márgenes, encabezados y pies, el título, autor y fecha, y otras muchas posibilidades.

En el preámbulo también se pueden definir nuevos comandos  $\text{\LaTeX}$ o redefinir los ya existentes.

### 2.1.3 Cuerpo

Contiene el texto del cuerpo del documento y tiene que ir dentro del entorno `document`. Suele empezar con el comando `\maketitle` si se desea empezar el documento con el título, autor y fecha que se hayan definido previamente en el preámbulo, y le sigue el comando `\tableofcontents` que introduce la tabla de contenidos en el documento. Finalmente iría el texto en sí con el contenido del documento.

## 2.2 Esqueleto básico para xelatex

Si se va a utilizar el compilador `xelatex` los paquetes del preámbulo cambian y deberían utilizarse los siguientes:

```
% Paquetes
\usepackage{fontspec}
\setmainfont{Times New Roman}
\usepackage{polyglossia}
\setdefaultlanguage{spanish}
```

El paquete `fontspec` permite definir las fuentes tipográfica que se desean utilizar en el documento final (por ejemplo `Times New Roman`), que debe estar instalada en el sistema donde se compile el documento, y el paquete `polyglossia` permite definir el idioma del documento.

---

<sup>1</sup>Una codificación de fuente es un mapeo de los códigos de caracteres a los glifos de la fuente que se utilizan para componer su salida.

En la Figura 2.1 se puede apreciar el documento final que se obtiene tras compilar el código anterior con `xelatex` en Overleaf.

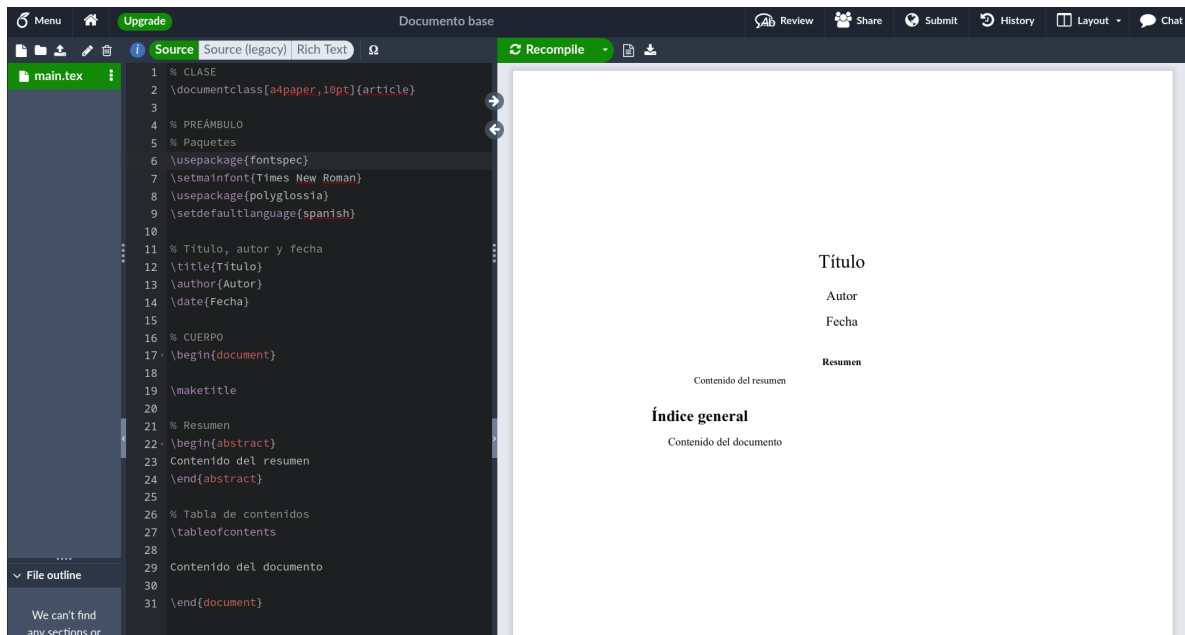


Figura 2.1: Estructura básica de un documento.

## 3 Secciones y párrafos

### 3.1 Secciones y subsecciones

Normalmente un documento extenso se dividirá en secciones y subsecciones (o incluso capítulos si se trata de un libro). Para definir las secciones de un documento se utilizan los siguientes comandos:

- `\chapter{Título del capítulo}`. Crea un nuevo capítulo con el título indicado y lo numera. Solo puede usarse cuando la clase del documento es `book`.
- `\section{Título de la sección}`. Crea una nueva sección con el título indicado y la numera.
- `\subsection{Título de la subsección}`. Crea una nueva subsección con el título indicado y la numera.
- `\subsubsection{Título de la subsubsección}`. Crea una nueva subsubsección con el título indicado y la numera.

Las secciones definidas con estos documentos aparecerán en la tabla de contenidos automáticamente.

Existen versiones alternativas de estos comandos añadiendo un asterisco (`\chapter*`, `\section*`, `\subsection*`, `\subsubsection*`) que crean encabezados de sección sin numerar y que tampoco aparecerán en la tabla de contenidos.

#### Ejemplo

```
\documentclass[a4paper, 10pt]{article}
...
% CUERPO
\begin{document}
\tableofcontents

\section{Sección primera}
Texto de la sección.

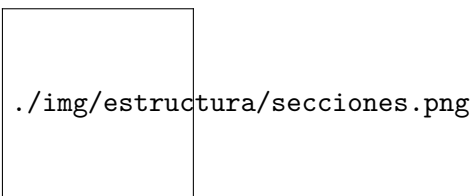
\subsection{Subsección primera}
Texto de la subsección.
```

```
% Encabezado de subsección sin numerar
\subsection*{Subsección segunda}
Texto de la subsección.

\subsection{Subsección tercera}
Texto de la subsección.

\section{Sección segunda}
Texto de la sección.
\end{document}
```

Salida



## 3.2 Párrafos y cambios de línea

Para crear un párrafo nuevo basta dejar una o más líneas en blanco.

Si se quiere hacer un cambio de línea dentro de un mismo párrafo, se utiliza el comando `\newline` o `\\`.

### Ejemplo

```
% CUERPO
\begin{document}
Este es el primer párrafo del documento, con un \\
cambio de línea.

Este es el segundo párrafo del documento. Obsérvese que cada vez que se comienza un párrafo
\end{document}
```

Salida

./img/estructura/parrafos.png

### 3.3 Justificación

Los párrafos se justifican por defecto a la izquierda y a la derecha.  $\text{\LaTeX}$  utiliza un algoritmo que permite partir las palabras al final de una línea para obtener párrafos con una buena estética (sin grandes espacios en blanco entre palabras). Pero también se pueden justificar solo a la izquierda, solo a la derecha o centrados entre los márgenes. Para ello se utilizan los siguientes entornos:

- `flushleft`: Justifica el texto a la izquierda.
- `flushright`: Justifica el texto a la derecha.
- `center`: Justifica el texto centrado entre los márgenes.

```
% CUERPO
```

```
\begin{document}
```

```
Este es el primer párrafo del documento, y aparece justificado a ambos lados (márgenes izq
```

```
\begin{flushleft}
```

```
Este es el segundo párrafo del documento y aparece justificado a la izquierda, es decir al
```

```
\end{flushleft}
```

```
\begin{flushright}
```

```
Este es el tercer párrafo del documento y aparece justificado a la derecha, es decir aline
```

```
\end{flushright}
```

```
\begin{center}
```

```
Este es el último párrafo del documento y aparece justificado en el centro entre los márg
```

```
\end{center}
```

```
\end{document}
```



## Salida

Este es el primer párrafo del documento, y aparece justificado a ambos lados (márgenes izquierdo y derecho) por defecto. Para que las líneas tengan la misma longitud, se utiliza un algoritmo que permite partir las palabras al final de una línea.

Este es el segundo párrafo del documento y aparece justificado a la izquierda, es decir alineado con el margen izquierdo del documento. Obsérvese que no todas las líneas acaban a la misma altura.

Este es el tercer párrafo del documento y aparece justificado a la derecha, es decir alineado con el margen derecho del documento. Obsérvese que no todas las líneas empiezan a la misma altura.

Este es el último párrafo del documento y aparece justificado en el centro entre los márgenes del documento. Obsérvese que ahora las líneas no empiezan ni terminan a la misma altura.

### Advertencia

El algoritmo para partir palabras al final de una línea funciona muy bien, pero si alguna vez divide mal una palabra, se puede indicar por dónde partir la palabra con el comando `\-` (por ejemplo `si\~la\~ba`).

## 4 Formateo básico

Existen multitud de comandos para dar formato al texto de un documento, pero en esta sección nos limitaremos a los más importantes.

### 4.1 Negrita, cursiva y subrayado

Para resaltar un texto habitualmente se utiliza negrita, cursiva o subrayado. Estos formatos se aplican con los siguientes comandos:

- `\textbf{...}`: Pone el texto en negrita.
- `\textit{...}`: Pone el texto en cursiva o itálica.
- `\emph{...}`: Enfatiza el texto cambiando de estilo (si estamos en un entorno de cursiva pasa a normal y si estamos en un entorno de texto normal pasa a cursiva).
- `\underline{...}`: Subraya el texto.

#### Ejemplo

```
% CUERPO
\begin{document}
Este texto está en \textbf{negrita}, este en \textit{cursiva} y este \underline{subrayado}.

\textit{Este texto está \emph{enfaticado}}.
\end{document}
```

Salida

Este texto está en **negrita**, este en *cursiva* y este subrayado.  
*Este texto está enfaticado.*

## 4.2 Familias de tipos de letra

Existen tres tipos de letra que se activan con los siguientes comandos:

- `\texrm{...}`: Texto normal (con **serif**). Es el tipo por defecto.
- `\textsf{...}`: Texto sin adornos (sin serif)
- `\texttt{...}`: Texto de máquina de escribir o monoespaciado (caracteres con la misma anchura).

### Ejemplo

```
% CUERPO
\begin{document}
Este texto es normal, \textsf{este es sin adornos}, \texttt{y este de máquina de escribir}
\end{document}
```

Salida

Este texto es normal, este es sin adornos, y este de máquina de escribir.

En el preámbulo del documento se puede seleccionar la fuente a utilizar para cada uno de ellos, especialmente con el paquete `fontspec` para compilar con `xelatex`. Para ello se utilizan los siguientes comandos:

- `\setromanfont{Fuente normal}`: Establece la fuente para el tipo de letra normal.
- `\setsansfont{Fuente sin adorno}`: Establece la fuente para el tipo de letra sin adorno.
- `\setmonofont{Fuente monoespaciada}`: Establece la fuente para el tipo de letra monoespaciado.

### Precaución

Las fuentes utilizadas en un documento deben estar previamente instaladas en el sistema operativo donde se compile el documento.

### Ejemplo

```
% PREÁMBULO
\usepackage{fontspec}
\setromanfont{Times New Roman}
\setsansfont{Arial}
```

```

\setmonofont{Courier New}

% CUERPO
\begin{document}
Este texto es normal, \textsf{este es sin adornos}, \texttt{y este de máquina de escribir}
\end{document}

```

Salida

Este texto es normal, **este es sin adornos**, y este de máquina de escribir.

## 4.3 Perfiles de letra

Para cada tipo de letra existen también varios perfiles que se activan con los siguientes comandos:

- `\textup{...}`: Activa el perfil recto. Es el perfil por defecto.
- `\textit{...}`: Activa el perfil de letra itálica.
- `\textsl{...}`: Activa el perfil inclinado.
- `\textsc{...}`: Activa el perfil de letra versalita (mayúsculas pequeñas).

### Ejemplo

```

% CUERPO
\begin{document}
Texto normal con perfil recto, \textit{itálica}, \textsl{inclinado} y \textsc{versalita}.

\textsf{Texto sin adorno con perfil recto, \textit{itálica}, \textsl{inclinado} y \textsc{versalita}.

\texttt{Texto monoespaciado con perfil recto, \textit{itálica}, \textsl{inclinado} y \textsc{versalita}.
\end{document}

```

## Salida

Texto normal con perfil recto, *itálica*, *inclinado* y VERSALITA.  
Texto sin adorno con perfil recto, *itálica*, *inclinado* y VERSALITA.  
Texto monoespaciado con perfil recto, *itálica*, *inclinado* y VERSALITA.

## 4.4 Tamaños de letra

A diferencia de otros procesadores donde el tamaño de la fuente se indica en puntos o pixels, en  $\text{\LaTeX}$  existen 10 tamaños predefinidos que se activan con los siguientes comandos, de menor a mayor tamaño:

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

Existen paquetes que permiten definir tamaños más pequeños o mayores pero no suelen ser necesarios en un documento normal.

### Ejemplo

```
% CUERPO
\begin{document}
\tiny{tiny}
\scriptsize{scriptsize}
\footnotesize{footnotesize}
\small{small}
\normalsize{normalsize}
\large{large}
\Large{Large}
\LARGE{LARGE}
\huge{huge}
```

```
\Huge{Huge}  
\end{document}
```

Salida

tiny scriptsize footnotesize small normalsize large Large LARGE huge

Huge

## 5 Listas

Existen tres tipos de listas, no ordenadas, ordenadas y descriptivas (en lugar de marcas o números los items de la lista están encabezados por texto), que se crean con los siguientes entornos:

- `itemize`: Crea una lista sin numerar.
- `enumerate`: Crea una lista enumerada.
- `description`: Crea una lista de tipo descripción.

Dentro de estos entornos, cada elemento de la lista debe empezar en una línea nueva con el comando `\item`. En el caso de las listas descriptivas, hay que proporcionar el texto del item de la lista como un argumento obligatorio.

### Ejemplos

Ejemplo de lista no ordenada.

```
% CUERPO
\begin{document}
Ejemplo de lista no ordenada:
\begin{itemize}
\item Este es un item.
\item Este es otro item.
\item Y otro item más.
\end{itemize}
\end{document}
```

## Salida

Ejemplo de lista no ordenada:

- Este es un ítem.
- Este es otro ítem.
- Y otro ítem más.

Ejemplo de lista ordenada.

```
% CUERPO
\begin{document}
Ejemplo de lista ordenada:
\begin{enumerate}
\item Primer ítem.
\item Segundo ítem.
\item Tercer ítem.
\end{enumerate}
\end{document}
```

## Salida

Ejemplo de lista ordenada:

1. Primer ítem.
2. Segundo ítem.
3. Tercer ítem.

Ejemplo de lista descriptiva.

```
% CUERPO
\begin{document}
Ejemplo de lista descriptiva:
\begin{description}
\item{\textbf{latex}} Genera documentos en formato dvi.
\end{description}
\end{document}
```



```

\item{\textbf{pdflatex}} Genera documentos en formato pdf.
\item{\textbf{xelatex}} Genera documentos en formato pdf que admiten codificación Unicode.
\end{description}
\end{document}

```

#### Salida

Ejemplo de lista descriptiva:

**latex** Genera documentos en formato dvi.

**pdflatex** Genera documentos en formato pdf.

**xelatex** Genera documentos en formato pdf que admiten codificación Unicode.

Se pueden crear sublistas anidando unos entornos dentro de otros.

#### Ejemplo

```

% CUERPO
\begin{document}
Ejemplo de listas anidadas:
\begin{enumerate}
\item Primer item.
    \begin{enumerate}
    \item Primer subitem.
    \item Segundo subitem.
    \end{enumerate}
\item Segundo item.
    \begin{itemize}
    \item Un item.
    \item Otro item.
    \end{itemize}
\end{enumerate}
\end{document}

```

## Salida

Ejemplo de listas anidadas:

1. Primer item.
  - (a) Primer subitem.
  - (b) Segundo subitem.
2. Segundo item.
  - Un item.
  - Otro item.

La indentación en el código fuente del ejemplo anterior no es obligatoria pero ayuda a ver mejor la estructura de anidamiento de entornos.

## 6 Tablas

Las tablas son uno de los elementos más complejos de  $\text{\LaTeX}$ , ya que, aunque es fácil crear una tabla sencilla, aplicarles un formato más avanzado con justificación de columnas, fusión de columnas o filas, márgenes de columnas, líneas de división, etc. suele ser bastante más difícil, aunque algunos entornos de edición facilitan la tarea. Existen multitud de paquetes para personalizar las tablas pero en esta sección solo veremos lo más básico.

Para crear una tabla se utiliza el entorno `tabular`. Este entorno tiene como argumento obligatorio el número de columnas de la tabla y su justificación, que se indica con una letra: `l` izquierda, `r` derecha y `c` centrada, (por ejemplo `lcr` indica tres columnas, la primera justificada a la izquierda, la segunda centrada y la tercera justificada a la derecha).

A continuación se introduce el contenido de la tabla, separando las filas con el comando de cambio de línea `\\` y dentro de cada línea separando las celdas con el comando `&`.

### Ejemplo

```
% CUERPO
\begin{document}
% Tabla con tres columnas, justificadas a la izquierda, centrada y derecha, respectivament
\begin{tabular}{lcr}
Nombre & Ciudad & Edad \\
María & Valencia & 22 \\
Juan & Madrid & 50 \\
Carmen & Barcelona & 35 \\
\end{tabular}
\end{document}
```

#### Salida

Nombre	Ciudad	Edad
María	Valencia	22
Juan	Madrid	50
Carmen	Barcelona	35

Para añadir líneas de división entre columnas, se introduce el carácter de barra vertical | entre las letras que definen la justificación de las columnas en el argumento obligatorio del entorno `tabular`. Mientras que para añadir líneas de división entre filas, se utiliza el comando `\hline` al principio de cada línea. Se insertarán tantas líneas divisorias como veces se introduzca el comando `\hline`.

### Ejemplo

```
% CUERPO
\begin{document}
% Tabla con líneas divisorias de filas y columnas.
\begin{tabular}{|l|c|r|}
\hline
Nombre & Ciudad & Edad \\
\hline
\hline
María & Valencia & 22 \\
\hline
Juan & Madrid & 50 \\
\hline
Carmen & Barcelona & 35 \\
\hline
\end{tabular}
\end{document}
```

#### Salida

Nombre	Ciudad	Edad
María	Valencia	22
Juan	Madrid	50
Carmen	Barcelona	35

El comando `\multicolumn{num}{col}{texto}` permite crear celdas que se extienden a lo largo de varias columnas, donde `num` es el número de columnas a ocupar, `col` es el carácter que define la justificación de la celda (l, r o c), y `texto` es el contenido de la celda.

También es posible utilizar el comando `\cline{n-m}` para dibujar líneas horizontales divisorias que no abarquen toda la fila, sino desde la columna `n` hasta la `m`.

### Ejemplo

```

% CUERPO
\begin{document}
\begin{tabular}{lrrcrr}
\hline
& \multicolumn{2}{c}{Enero} & & \multicolumn{2}{c}{Febrero}\\
\cline{2-3}\cline{5-6}
Ciudad & Ingresos & Gastos & & Ingresos & Gastos\\
\hline
Madrid & 2500 & 1750 & & 2600 & 1800\\
Barcelona & 2250 & 1500 & & 2400 & 1650\\
\hline
\end{tabular}
\end{document}

```

Salida

Ciudad	Enero		Febrero	
	Ingresos	Gastos	Ingresos	Gastos
Madrid	2500	1750	2600	1800
Barcelona	2250	1500	2400	1650

## 7 Imágenes

Para incluir una imagen o figura en un documento, además de disponer de la imagen del fichero en un formato gráfico adecuado, es necesario cargar en el preámbulo el paquete **graphicx**. Este paquete permite gestionar imágenes en los formatos gráficos **jpg**, **png**, **tiff**, **eps** y **pdf** (los tres primeros son formatos de mapas de bits y los dos últimos vectoriales).

Una vez cargado el paquete, para insertar una imagen en el documento basta con utilizar el comando `\includegraphics[opiones]{fichero}`. Este comando tiene como argumento obligatorio es el nombre del fichero con la imagen (incluyendo la ruta en el sistema de ficheros local) y los siguientes argumentos opcionales para modificar el aspecto de la imagen:

- **height**: Indica la altura de la imagen. Escala la imagen hasta esa altura.
- **width**: Indica la anchura de la imagen. Escala la imagen hasta esa anchura.
- **scale**: Factor de escalado de la imagen de 0 a 1.
- **angle**: Ángulo de rotación de la imagen. Rota la imagen en el sentido de las agujas del reloj los grados indicados.

### Ejemplo

```
% PREÁMBULO
\usepackage{graphicx}
% CUERPO
\begin{document}
Ejemplo de imagen en línea
\includegraphics{img/logo-aprendeconalf.png},
escalada
\includegraphics[height=1cm]{img/logo-aprendeconalf.png},
y rotada
\includegraphics[angle=90]{img/logo-aprendeconalf.png}

Ejemplo de imagen centrada:

\begin{center}
\includegraphics{img/logo-aprendeconalf.png}
\end{center}
\end{document}
```

## Salida

Ejemplo de imagen en línea , escalada , y rotada   
Ejemplo de imagen centrada:



## 8 Fórmulas matemáticas

La escritura de fórmulas matemáticas es uno de los puntos fuertes de  $\text{\LaTeX}$ , y es por ello que se utiliza tanto para la creación de documentos científicos o técnicos con contenido matemático.

Para escribir una fórmula es necesario cambiar al modo matemático. Existen distintas formas de activar el modo matemático:

- $\text{\$}$ : Activa el modo matemático en línea, es decir, las fórmulas aparecerán en la misma línea que el texto que las rodea. Para desactivar este modo hay que volver a escribir  $\text{\$}$ .
- $\text{\$\$}$ : Activa el modo matemático *display* (desplegado), de manera que las fórmulas aparecen en una línea aparte.
- El entorno `equation` también activa el modo matemático *display* pero además asigna un número a la ecuación, para poder referenciarla en otras partes del documento.

### Ejemplo

```
% CUERPO
\begin{document}
Ejemplo de fórmula en línea  $x+y=0$  $.

Ejemplo de fórmula desplegada


$$x+y=0$$


Ejemplo de fórmula con el entorno \texttt{equation}

\begin{equation}
x+y=0
\end{equation}
\end{document}
```



## Salida

Ejemplo de fórmula en línea  $x + y = 0$ .

Ejemplo de fórmula desplegada

$$x + y = 0$$

Ejemplo de fórmula con el entorno `equation`

$$x + y = 0 \tag{1}$$

## 8.1 Símbolos matemáticos

Existe una infinidad de símbolos matemáticos que escriben mediante comandos. A continuación se muestran los más habituales.<sup>1</sup>

### 8.1.1 Letras griegas

Para escribir letras griegas se utilizan los siguientes comandos:

#### Minúsculas

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

#### Mayúsculas

<code>\Gamma</code> $\Gamma$	<code>\Lambda</code> $\Lambda$	<code>\Sigma</code> $\Sigma$	<code>\Psi</code> $\Psi$
<code>\Delta</code> $\Delta$	<code>\Xi</code> $\Xi$	<code>\Upsilon</code> $\Upsilon$	<code>\Omega</code> $\Omega$

<sup>1</sup>Para un listado más exhaustivo de los símbolos matemáticos de L<sup>A</sup>T<sub>E</sub>X, puede consultarse el documento [The Great, Big List of L<sup>A</sup>T<sub>E</sub>X Symbols](#).

<code>\Theta</code> $\Theta$	<code>\Pi</code> $\Pi$	<code>\Phi</code> $\Phi$
------------------------------	------------------------	--------------------------

### 8.1.2 Operadores aritméticos

<code>+</code> $+$	<code>-</code> $-$	<code>\times</code> $\times$	<code>\cdot</code> $\cdot$
<code>/</code> $/$	<code>\div</code> $\div$	<code>\sqrt{...}</code> $\sqrt{\quad}$	<code>\pm</code> $\pm$

### 8.1.3 Relaciones

<code>=</code> $=$	<code>\neq</code> $\neq$	<code>&lt;</code> $<$	<code>\leq</code> $\leq$
<code>&gt;</code> $>$	<code>\geq</code> $\geq$	<code>\approx</code> $\approx$	<code>\sim</code> $\sim$
<code>\equiv</code> $\equiv$	<code>\in</code> $\in$	<code>\notin</code> $\notin$	<code>\subset</code> $\subset$
<code>\not\subset</code> $\not\subset$	<code>\subseteq</code> $\subseteq$	<code>\subsetneq</code> $\subsetneq$	

### 8.1.4 Operadores binarios

<code>\cup</code> $\cup$	<code>\cap</code> $\cap$	<code>\setminus</code> $\setminus$	<code>\circ</code> $\circ$
--------------------------	--------------------------	------------------------------------	----------------------------

### 8.1.5 Lógica

<code>\exists</code> $\exists$	<code>\forall</code> $\forall$	<code>\neg</code> $\neg$	<code>\lor</code> $\lor$	<code>\land</code> $\land$
--------------------------------	--------------------------------	--------------------------	--------------------------	----------------------------

### 8.1.6 Conjuntos

<code>\emptyset</code> $\emptyset$	<code>\mathbb{N}</code> $\mathbb{N}$	<code>\mathbb{Z}</code> $\mathbb{Z}$	<code>\mathbb{Q}</code> $\mathbb{Q}$
<code>\mathbb{R}</code> $\mathbb{R}$	<code>\mathbb{C}</code> $\mathbb{C}$		

### 8.1.7 Flechas

<code>\rightarrow</code> $\rightarrow$	<code>\Rightarrow</code> $\Rightarrow$	<code>\longrightarrow</code> $\longrightarrow$	<code>\Longrightarrow</code> $\Longrightarrow$
<code>\leftarrow</code> $\leftarrow$	<code>\Leftarrow</code> $\Leftarrow$	<code>\longleftarrow</code> $\longleftarrow$	<code>\Longleftarrow</code> $\Longleftarrow$

---

<code>\leftrightharpoon</code>	$\leftrightarrow$	<code>\Leftrightarrow</code>	$\Leftrightarrow$	<code>\longleftarrow</code>	$\longleftarrow$	<code>\Longleftarrow</code>	$\Longleftarrow$
<code>\uparrow</code>	$\uparrow$	<code>\Uparrow</code>	$\Uparrow$	<code>\downarrow</code>	$\downarrow$	<code>\Downarrow</code>	$\Downarrow$
<code>\updownarrow</code>	$\updownarrow$	<code>\Updownarrow</code>	$\Updownarrow$				

---

### 8.1.8 Otros s mbolos

---

<code>\infty</code>	$\infty$	<code>\partial</code>	$\partial$	<code>\nabla</code>	$\nabla$
---------------------	----------	-----------------------	------------	---------------------	----------

---

### 8.1.9 Funciones

---

<code>\sin</code>	$\sin$	<code>\arcsin</code>	$\arcsin$	<code>\csc</code>	$\csc$	<code>\operatorname{arccsc}</code>	$\operatorname{arccsc}$
<code>\cos</code>	$\cos$	<code>\arccos</code>	$\arccos$	<code>\sec</code>	$\sec$	<code>\operatorname{arcsec}</code>	$\operatorname{arcsec}$
<code>\tan</code>	$\tan$	<code>\arctan</code>	$\arctan$	<code>\cot</code>	$\cot$	<code>\operatorname{arccot}</code>	$\operatorname{arccot}$
<code>\exp</code>	$\exp$	<code>\log</code>	$\log$	<code>\ln</code>	$\ln$		

---

Es posible declarar nuevos operadores o funciones cargando el paquete `amsmath` con el comando `DeclareMathOperator{comando}{texto}`. Por ejemplo, para obtener la versi n de la funci n seno en espa ol se puede definir `DeclareMathOperator{\sen}{sen}` en el pre mbulo y luego utilizar el comando `\sen` en el cuerpo para obtener la funci n seno en espa ol.

Otro paquete que incorpora a n m s s mbolos es `amssymb`.

## 8.2 Sub ndices y super ndices

Para poner sub ndices se utiliza el comando `_` y para super ndices `^`. Si el sub ndice o super ndice afecta a m s de un car cter, hay que ponerlos entre llaves.

### Ejemplo

```
% CUERPO
\begin{document}
Ejemplo de f rmula con sub ndices

$$
```

```
x_i+y_j=0
$$
```

Ejemplo de fórmula con superíndices

```
$$
x^2+y^2=0
$$
```

Ejemplo de fórmula con subíndices y superíndices

```
$$
x_i^2+y_j^2=0
$$
```

```
\end{document}
```

#### Salida

Ejemplo de fórmula con subíndices

$$x_i + y_j = 0$$

Ejemplo de fórmula con superíndices

$$x^2 + y^2 = 0$$

Ejemplo de fórmula con subíndices y superíndices

$$x_i^2 + y_j^2 = 0$$

Se pueden escribir subíndices de subíndices o superíndices de superíndices anidando los comandos.

## 8.3 Fracciones

Para escribir fracciones simples en línea se puede usar el operador aritmético / (por ejemplo  $3/4$ ), pero para fracciones más complejas o fracciones en modo display, conviene utilizar el comando `\frac{num}{den}`, donde **num** es el numerador y **den** el denominador.

A su vez, se pueden escribir más fracciones en el numerador o el denominador, anidando este comando.

### Ejemplo

```
% CUERPO
\begin{document}
Ejemplo de fracción en línea $\frac{x+2}{x^2-2x+1}$.

Ejemplo de fracción en modo desplegado

$$
\frac{\frac{x}{2}+\frac{2}{3}}{x^2-2x+1}
$$

\end{document}
```

#### Salida

Ejemplo de fracción en línea  $\frac{x+2}{x^2-2x+1}$ .

Ejemplo de fracción en modo desplegado

$$\frac{\frac{x}{2} + \frac{2}{3}}{x^2 - 2x + 1}$$

## 8.4 Sumatorios, productorios e integrales

Para escribir sumatorios se utiliza el comando `\sum_{sub}^{sup}`, donde *sub* es el subíndice que indica el inicio de la suma y *sup* es el superíndice que indica el final de la suma. Si se quieren omitir los índices de inicio y final de la suma, basta con el comando `\sum`.

De manera análoga, para escribir productorios se utiliza el comando `\prod_{sub}^{sup}`, donde `sub` es el subíndice que indica el inicio del producto y `sup` es el superíndice que indica el final del producto. Si se quieren omitir los índices de inicio y final del producto, basta con el comando `\prod`.

### Ejemplo

```
% CUERPO
\begin{document}
Ejemplo de sumatorio
$$
\sum_{i=1}^{\infty} x^i
$$

Ejemplo de productorio
$$
\prod_{i=1}^n i
$$

\end{document}
```

#### Salida

Ejemplo de sumatorio

$$\sum_{i=1}^{\infty} x^i$$

Ejemplo de productorio

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

Del mismo modo, para escribir integrales definidas se utiliza el comando `\int_{sub}^{sup}`, donde `sub` es el subíndice que indica el inicio de la integral y `sup` es el superíndice que indica el final de la integral. Para integrales indefinidas, basta con el comando `\int`.

### Ejemplo

```
% CUERPO
\begin{document}
Ejemplo de integral definida
$$
\int_a^b f(x) dx
$$

Ejemplo de integral indefinida
$$
\int f(x) dx
$$

\end{document}
```

Salida

Ejemplo de integral definida

$$\int_a^b f(x) dx$$

Ejemplo de integral indefinida

$$\int f(x) dx$$

## 8.5 Sombreros

Es posible poner símbolos encima de otros símbolos, más conocidos como *sombreros*. Los siguientes comandos sirven para poner distintos tipos de sombreros:

- `\bar{...}`: Línea horizontal para un carácter.
- `\overline{...}`: Línea horizontal para varios caracteres.
- `\hat{...}`: Ángulo para un carácter.
- `\widehat{...}`: Ángulo para varios caracteres.

- `\vec{...}`: Flecha para un carácter.
- `\overrightarrow{...}`: Flecha para varios caracteres.

### Ejemplo

```
% CUERPO
\begin{document}
Ejemplos de sombreros:  $\overline{xy}$ ,  $\hat{a}$ ,  $\widehat{abc}$ ,  $\vec{u}$ .
\end{document}
```

Salida

Ejemplos de sombreros:  $\overline{xy}$ ,  $\hat{a}$ ,  $\widehat{abc}$ ,  $\vec{u}$ .

## 8.6 Matrices

Las matrices se crean de manera similar a como se crean las tablas (ver Capítulo 6), pero utilizando el entorno `array` en lugar del entorno `tabular`. Para encerrar la matriz entre paréntesis se pone el comando `\left(` antes del entorno y el comando `\right)` después.

### Ejemplo

```
% CUERPO
\begin{document}
Ejemplo de matriz
$$
\left(
\begin{array}{rrr}
1 & 2 & 3 \\
x & y & z
\end{array}
\right)
$$

\end{document}
```



## Salida

Ejemplo de matriz

$$\begin{pmatrix} 1 & 2 & 3 \\ x & y & z \end{pmatrix}$$

El paquete `amsmath` incorpora varios entornos más específicos para matrices donde no es necesario especificar el número de columnas, y tampoco los delimitadores:

- `matrix`: Matriz sin delimitadores (equivalente al entorno `array`)
- `pmatrix`: Matriz encerrada entre paréntesis.
- `vmatrix`: Matriz encerrada entre barras verticales (por ejemplo para determinantes).
- `Vmatrix`: Matriz encerrada entre dobles barras verticales.
- `bmatrix`: Matriz encerrada entre corchetes.
- `Bmatrix`: Matriz encerrada entre llaves.

## Ejemplo

```
% PREÁMBULO
\usepackage{amsmath}
% CUERPO
\begin{document}
Ejemplo de determinante

$$
\begin{vmatrix}
1 & x & \alpha \\
2 & y & \beta \\
3 & z & \gamma
\end{vmatrix}
$$

\end{document}
```

## Salida

Ejemplo de determinante

$$\begin{vmatrix} 1 & x & \alpha \\ 2 & y & \beta \\ 3 & z & \gamma \end{vmatrix}$$

## 8.7 Teoremas

Para crear definiciones, teoremas, proposiciones, corolarios y otros tipos de enunciados se debe cargar el paquete `amsthm` y definir los tipos de enunciados en el preámbulo con el comando `\newtheorem{entorno}{texto}`, donde `entorno` es el nombre del entorno y `texto` el texto que aparecerá en el documento final como encabezado del enunciado.

Estos entornos admiten como argumento opcional un texto que se utiliza para dar nombre al enunciado.

Los enunciados definidos con este comando aparecen por defecto numerados para poder referenciarlos en otras partes del documento, pero se pueden definir entornos no numerados con la variante del comando anterior `\newtheorem*{entorno}{texto}`

Para demostraciones se puede utilizar el entorno `proof`.

### Ejemplo

```
% PREÁMBULO
\usepackage{amsmath}
\DeclareMathOperator{\sen}{sen}
\usepackage{amsthm}
\newtheorem{midef}{Definición}
\newtheorem{teo}{Teorema}

% CUERPO
\begin{document}

\begin{midef}
Dado un triángulo rectángulo de catetos  $a$ ,  $b$  e hipotenusa  $c$ , se define el seno del ángulo  $\alpha$  como:

$$\sin(\alpha) = \frac{b}{c}.$$

\end{midef}
```

\$\$

`\end{midef}`

`\begin{teo}`

Para cualquier ángulo  $\alpha$  se cumple  $\sin^2(\alpha) + \cos^2(\alpha) = 1$ .

`\end{teo}`

`\begin{proof}`

Es una consecuencia directa del teorema de Pitágoras.

`\end{proof}`

`\end{document}`

#### Salida

**Definición 1.** Dado un triángulo rectángulo de catetos  $a$ ,  $b$  e hipotenusa  $c$ , se define el seno del ángulo  $\alpha$  opuesto al cateto  $b$  como

$$\operatorname{sen} \alpha = \frac{b}{c}.$$

**Teorema 1.** Para cualquier ángulo  $\alpha$  se cumple  $\operatorname{sen}(\alpha)^2 + \cos(\alpha)^2 = 1$ .

*Demostración.* Es una consecuencia directa del teorema de Pitágoras.  $\square$

#### Tip

Se recomienda cargar los paquetes `amsmath`, `amssymb` y `amsthm` para documentos extensos o con muchas fórmulas matemáticas.

## 9 Entornos flotantes

Hay determinados contenidos, como por ejemplo las tablas o las imágenes que son bloques indivisibles, de manera que cuando no hay espacio suficiente en la página para encajarlos, pasan a colocarse en la siguiente página, dejando en la página anterior un espacio vertical vacío poco estético.

La solución consiste en incluir estos contenidos en un entorno flotante, que se ubicará automáticamente sin dejar espacios vacíos. Como estos contenidos pueden aparecer lejos de su posición en el código fuente, para que no estén descontextualizados suelen llevar asociada una leyenda.

Existen dos entornos flotantes, para figuras y tablas.

### 9.1 Entorno flotante para figuras

El entorno flotante para figuras es `figure` tiene el siguiente esqueleto.

```
\begin{figure}[posición]
    Código de las imágenes
\label{etiqueta}
\caption{leyenda}
\end{figure}
```

El argumento opcional indica la preferencia de ubicación de la figura en la página (`h` en el lugar en el que aparece en el código fuente, `t` arriba, `b` abajo).  $\text{\LaTeX}$  intentará ubicar la figura en esa posición salvo que no sea posible.

Las figuras flotantes se numeran automáticamente y el comando `\label{...}` asigna una etiqueta al entorno flotante para poder referenciarlo desde otras partes del documento. Por su parte, el comando `\caption{...}` crea la leyenda de la figura.

```
% PREÁMBULO
\usepackage{graphicx}
% CUERPO
\begin{document}
Ejemplo de imagen flotante. Como se puede apreciar la imagen aparece al principio de la pá
```

```

\begin{figure}[t]
\begin{center}
\includegraphics{img/logo-aprendeconalf.png}
\end{center}
\label{img-1}
\caption{Logotipo del sitio web AprendeconAlf.}
\end{figure}
\end{document}

```

Salida



Figura 1: Logotipo del sitio web AprendeconAlf.

Ejemplo de imagen flotante. Como se puede apreciar la imagen aparece al principio de la página aunque va después de este párrafo en el código fuente.

Para incluir el listado de figuras de un documento en cualquier parte se utiliza el comando `\listoffigures`.

## 9.2 Entorno flotante para tablas

El entorno flotante para tablas es `table` y su esqueleto es muy parecido al del entorno para figuras.

```

\begin{table}[posición]
  Código de la tabla
\label{etiqueta}
\caption{leyenda}
\end{table}

```

Las tablas, al igual que las figuras, se enumeran automáticamente y pueden referenciarse después asignándoles una etiqueta con el comando `\label{...}`.

```
% CUERPO
\begin{document}
Ejemplo de tabla flotante. Como se puede apreciar la tabla aparece al principio de la página

\begin{table}[t]
\begin{center}
\begin{tabular}{|l|c|r|}
\hline
Nombre & Ciudad & Edad \\
\hline
\hline
María & Valencia & 22 \\
\hline
Juan & Madrid & 50 \\
\hline
Carmen & Barcelona & 35 \\
\hline
\end{tabular}
\end{center}
\label{img-1}
\caption{Tabla de clientes de una empresa.}
\end{table}
\end{document}
```

Salida

Nombre	Ciudad	Edad
María	Valencia	22
Juan	Madrid	50
Carmen	Barcelona	35

Cuadro 1: Tabla de clientes de una empresa.

Ejemplo de tabla flotante. Como se puede apreciar la tabla aparece al principio de la página aunque va después de este párrafo en el código fuente.

Para incluir el listado de figuras de un documento en cualquier parte se utiliza el comando `\listoftables`.

# 10 Referencias cruzadas y notas a pie

Otro de los puntos fuertes de  $\text{\LaTeX}$  es la gestión de las referencias cruzadas, es decir, referencias a otras partes del documento, así como las notas a pie de página.

## 10.1 Referencias cruzadas

Como se ha visto, muchos elementos de un documento están enumerados: capítulos, secciones, figuras, tablas, ecuaciones, teoremas, páginas, etc. Para poder referenciarlos, cada elemento debe tener asignada una etiqueta única. Para asignar una etiqueta a cualquier elemento numerado se utiliza el comando `\label{etiqueta}`. Este comando debe ubicarse justo antes o después del elemento que quiere etiquetar.

Posteriormente, para hacer una referencia al elemento en otra parte del documento se utiliza el comando `\ref{etiqueta}`.

### Ejemplo

```
% PREÁMBULO
\usepackage{amsmath}
\DeclareMathOperator{\sen}{sen}
\usepackage{amsthm}
\newtheorem{teo}{Teorema}

% CUERPO
\begin{document}
\begin{teo}\label{teo-trigo}
Para cualquier ángulo
 $\alpha$  se cumple
\begin{equation}\label{eq-trigo}
\sen(\alpha)^2 + \cos(\alpha)^2 = 1.
\end{equation}
\end{teo}
```

```
Ejemplo de referencia cruzada. La ecuación \ref{eq-trigo} del teorema \ref{teo-trigo} es u
\end{document}
```



Salida

**Teorema 1.** *Para cualquier ángulo  $\alpha$  se cumple*

$$\operatorname{sen}(\alpha)^2 + \operatorname{cos}(\alpha)^2 = 1. \quad (1)$$

Ejemplo de referencia cruzada. La ecuación 1 del teorema 1 es una ecuación básica en trigonometría.

## 10.2 Notas a pie de página

Para insertar una nota a pié de página se utiliza el comando `\footnote{...}`.

**Ejemplo**

```
% CUERPO
\begin{document}
El logotipo de latex es  $\LaTeX$ .\footnote{Fue creado por Leslie Lamport.}
\end{document}
```

Salida

El logotipo de latex es  $\LaTeX$ .<sup>1</sup>

---

<sup>1</sup>Fue creado por Leslie Lamport.

# Bibliografía

- Cascales, B. y otros (2003). *El libro de Latex*. Pearson Educación.
- Cascales, B. y otros (2000). *LaTeX: una imprenta en sus manos*. Aula Documental de Investigación.
- Lamport, Leslie (1994). *LaTeX: A Document Preparation System*, 2nd Edition. 2nd ed. Addison-Wesley Professional. .
- Mittelbach, F. et al. (2004). *LaTeX Companion, The, 2nd Edition*. 2nd ed. Addison-Wesley Professional.