

Tutorial de L^AT_EX

Carlos Rodríguez Domínguez

Mayo 2021

Índice general

1. Sintáxis básica	1
1.1. Comentarios	1
1.2. Comandos	1
1.2.1. Comandos personalizados	2
1.3. Símbolos reservados y caracteres especiales	2
1.4. Preámbulo	2
1.5. Tipos de documentos	3
1.6. Estructura de un documento	3
1.7. Paquetes	4
1.8. Estilos básicos	4
1.9. Tamaños y tipos de fuentes	4
1.10. Colores	5
1.10.1. Texto	5
1.10.2. Fondo de páginas y color por defecto	6
1.11. Alineación de textos	6
1.12. Notas al pie y al margen	6
1.13. Listas	6
1.13.1. Estilos en listas no numeradas	7
1.13.2. Estilos en listas numeradas	8
1.14. División en múltiples archivos	9
2. Tablas, figuras, ecuaciones e índices	11
2.1. Figuras	11
2.2. Tablas	13
2.2.1. Tamaños de columnas	13
2.2.2. Combinación de filas y columnas	14
2.2.3. Importación desde CSV	14
2.3. Listados de código	16

2.4.	Ecuaciones	18
2.5.	Posicionamiento	19
2.6.	Leyendas	22
2.6.1.	Figuras o tablas	22
2.6.2.	Subfiguras	23
2.6.3.	Código	23
2.7.	Etiquetas y referencias cruzadas	24
2.8.	Índices	25
3.	Gestión bibliográfica	29
3.1.	Entradas manuales y citas	29
3.2.	Bases de datos bibliográficas externas y estilos automáticos . .	29

Notas iniciales

El objetivo de este documento es demostrar las funcionalidades más básicas y comunes de \LaTeX .

Este documento debe leerse compaginando la lectura del código fuente con el PDF generado por \LaTeX . Es muy recomendable mantener en pantalla ambos contenidos en todo momento, y tratar de entender cómo se generan los contenidos del documento PDF a partir de los comandos de \LaTeX .

Se asume que el lector de este documento ya posee una instalación de \LaTeX y un editor de dicho lenguaje.

Capítulo 1

Sintáxis básica

\LaTeX permite escribir textos de manera cómoda y sencilla. Nos va a ayudar a darle un estilo profesional a todo lo que escribamos, y a autogenerar índices, capítulos, secciones, etc.

Al principio \LaTeX es un poco abrumador, así que trataremos de aprenderlo paso a paso.

Para ello, leeremos este documento con el código a su lado. De esta manera observaremos cómo se podría escribir un documento similar a este.

1.1. Comentarios

Dentro de nuestro código podemos dejar indicaciones para recordar ciertos detalles o dar instrucciones a nuestros compañeros. Los comentarios jamás se visualizarán en el documento final. Los comentarios debemos comenzarlos por el símbolo `%`.

1.2. Comandos

Los comandos son instrucciones para \LaTeX que permiten formatear nuestro documento como deseemos. Siempre comienzan por la letra `\`.

Los comandos tienen parámetros y opciones. Los parámetros son aquellos contenidos que toma el comando para poder producir los resultados que esperamos. Se ponen entre `{ }`.

Las opciones son parámetros no obligatorios que habitualmente alteran la manera de actuar del comando principal. Se ponen entre `[]`. Por ejemplo,

permiten cambiar el tamaño por defecto de una figura (se verá más adelante).

1.2.1. Comandos personalizados

Se pueden crear nuevos comandos en el preámbulo mediante el comando `\newcommand{\nombre}[número. parámetros]{definición}`.

De esta forma, si pusiésemos `\newcommand{\hello}[1]{Hola {#1}}` podríamos usar el comando `\hello{Mundo}`, cuyo resultado sería “Hola Mundo”.

Asimismo, podemos cambiar los comandos de \LaTeX de cualquier paquete usando el comando `\renewcommand`, con la misma sintáxis que `\newcommand`.

1.3. Símbolos reservados y caracteres especiales

Para usar dentro de nuestros textos los símbolos que utiliza \LaTeX para dar estilo a los textos o especificar el preámbulo, que son: `# $ % & { } _ ^ \`, debemos anteponer en nuestro código a dichos símbolos el símbolo `\`. El propio símbolo `\` es un tanto especial, y hay que escribirlo mediante el comando `\textbackslash`.

Por otro lado, si deseamos insertar caracteres especiales lo tendremos que hacer mediante comandos. Un amplio listado de dichos comandos está disponible en: https://en.wikibooks.org/wiki/LaTeX/Special_Characters.

1.4. Preámbulo

Los documentos \LaTeX incluyen códigos para que se puedan generar los documentos finales adecuadamente. Todo lo que se escribe en el código antes de `\begin{document}` se le conoce como el preámbulo.

El preámbulo incluye información sobre los paquetes que necesitamos usar para generar el documento final, el tipo de documento que estamos escribiendo, tamaño de letra, estilo básico, etc.

Además, en el preámbulo se especifica el título, autor y fecha del documento.

1.5. Tipos de documentos

L^AT_EX es capaz de aplicar estilos predefinidos a diversos tipos de documentos: libros, artículos, etc. Dichos estilos se aplican en el preámbulo, dentro del comando `\documentclass`.

Entre los estilos de los que disponemos por defecto (se recomienda probarlos) están: *article*, *report*, *book*, etc. L^AT_EX permite definir nuestros propios estilos (muy avanzado), o bien incluir estilos de terceros (muy habitual en congresos/revistas).

Asimismo, el comando `\documentclass` permite especificar el tamaño del papel (*a4paper*, *a5paper*, *executivepaper*, etc.), de la tipografía por defecto, el número de columnas (*onecolumn*, *twocolumn*), y un largo etcétera de opciones avanzadas.

1.6. Estructura de un documento

Los documentos se estructuran en capítulos (`\chapter{nombre}`), secciones (`\section{nombre}`), subsecciones (`\subsection{nombre}`), sub-subsecciones (`\subsubsection{nombre}`), subsubsub...section (`\subsubsub...section{nombre}`). Si tras cualquiera de estos elementos ponemos un `*` (e.g.: *chapter*{Nombre capítulo}*), entonces dicho elemento no se numerará y no se incluirá por defecto en ningún índice.

Según el tipo de documento, puede que algunos de estos elementos estructurales no estén disponibles, y en cambio estén disponibles otros en su lugar.

Por ejemplo, el tipo de documento *beamer* permite crear con L^AT_EX presentaciones de diapositivas. En este caso no se aceptaría la división por capítulos, y en su lugar se permitiría la definición de diapositivas entre los comandos `\begin{frame} ... \end{frame}`.

En el tipo de documento *book* se permite también la creación de anexos como capítulos especiales definidos entre los comandos `\begin{appendices} ... \end{appendices}`.

Por otro lado, en el tipo *book* podemos especificar qué partes de nuestro documento están orientados al título, abstract, índices, etc., al contenido principal y a la bibliografía, glosarios, etc. Para ello, usaremos, respectivamente el comando `\frontmatter`, `\mainmatter` y `\backmatter` justo antes del comienzo de cada uno de estos apartados de nuestro documento. Estos co-

mandos permiten que los números de páginas y el estilo general de las páginas sean diferentes según la parte del documento que estemos especificando.

Finalmente, podremos siempre usar el comando `\clearpage` para forzar un salto de página.

1.7. Paquetes

Antes hemos mencionado los “paquetes”. Un paquete es un archivo de \LaTeX que incluye nuevos estilos, comandos, opciones, etc. para dar un formato apropiado a nuestros documentos. Hay multitud de dichos paquetes, y es imprescindible conocer algunos de ellos para poder escribir un documento.

Un paquete se incluye en el preámbulo mediante la palabra `\usepackage`. Podemos observar en el código de este documento los paquetes más básicos que se usan en casi cualquier documento \LaTeX .

El nombre del paquete será el indicado entre `{ }`. Entre `[]` se introducen las opciones para configurar dicho paquete.

Los, literalmente, cientos de paquetes disponibles en \LaTeX se pueden consultar en: <https://ctan.org/pkg>.

Es bueno que al incluir un paquete dejemos un pequeño comentario en nuestro código para indicar para qué sirve dicho paquete.

1.8. Estilos básicos

En \LaTeX podemos, al igual que en Word, escribir en **negrita**, *cursiva* o subrayar.

Para crear párrafos formateados adecuadamente debemos siempre recordar dar dos intros en nuestro código.

Con \LaTeX es bueno atender a los detalles de estilo para conseguir unos resultados muy *profesionales*. La forma alternativa de introducir las dobles comillas “” es un ejemplo de ello.

1.9. Tamaños y tipos de fuentes

En el preámbulo podemos especificar el tamaño por defecto de la fuente para todo el documento (comando `documentclass`). No obstante, podemos especificar cualquiera de los siguientes tamaños por defecto para nuestros

textos (una vez que ponemos un tamaño de letra, continuará dicho tamaño hasta que no pongamos otro):

Texto, Texto, Texto, Texto, Texto, Texto (por defecto), Texto, Texto, Texto, Texto

Asimismo, podemos especificar tamaños personalizados de fuente y de interlineado:

Texto

Igualmente podemos especificar el tipo de fuente:

Esta frase tiene una tipografía *serif* (por defecto)

Esta frase tiene una tipografía *sans serif*

Esta frase tiene una tipografía *typewriter*

O bien especificarla para todo nuestro documento en el preámbulo mediante el comando:

`\renewcommand{\familydefault}{\sfdefault}`

Los tipos de fuentes disponibles para el preámbulo son: `\rmdefault` (serif, por defecto), `\sfdefault` (sans serif), `\ttdefault` (typewriter)

Mediante paquetes adicionales se dispone de una gran variedad de tipografías (tantas como en cualquier procesador de textos).

1.10. Colores

1.10.1. Texto

Al incluir el paquete “xcolor” en el preámbulo podremos colorear nuestro texto. Para ello usaremos el comando `\textcolor{nombre color}`, siendo el nombre del color cualquiera de los disponibles de manera predeterminada o cualquier definido por nosotros en el preámbulo del documento.

Veamos un ejemplo:

Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin.

Podemos consultar el listado completo de colores básicos y avanzados en: https://es.overleaf.com/learn/latex/Using_colours_in_LaTeX#Reference_guide.

Si algún color no está disponible en dicho listado, o deseamos especificar uno muy concreto, podemos usar el comando `\definecolor{nombre nuevo color}{formato (e.g.: RGB)}{valores (e.g.: 219, 48, 122)}` en el preámbulo del documento. A partir de ese momento podremos usar ese nombre de color para especificar dicho color en todo nuestro documento.

1.10.2. Fondo de páginas y color por defecto

El color de fondo se especifica mediante el comando `\pagecolor{nombre color}` y el color del texto por defecto mediante `\color{nombre color}`.

1.11. Alineación de textos

El texto lo podemos escribir alineado a la izquierda,

centrado

o alineado a la derecha,

1.12. Notas al pie y al margen

\LaTeX permite crear notas al pie o al margen de página de manera muy simple, usando los comandos `\footnote{nota al pie}` y `\marginpar{nota al margen}`.

Por ejemplo, este párrafo tiene asociado una nota al pie de página¹.

Este otro tiene asociado una nota al margen (podemos observar como las notas al margen no están numeradas).

Las notas al margen de página en \LaTeX son muy sencillas de incluir

1.13. Listas

Existen dos tipos de listas: Numeradas y no numeradas. Un ejemplo de ambas es el siguiente:

1. Primero

2. Segundo

¹Las notas al pie de página en \LaTeX son muy sencillas de incluir

- Primero
- Segundo

Además, se pueden anidar:

- Primero
 1. A
 2. B
- Segundo
 1. A
 2. B

1.13.1. Estilos en listas no numeradas

Podemos personalizar los estilos de las listas mediante mediante el siguiente código en el preámbulo (se cambiaría el estilo de todas las listas de todo el documento) o justo antes de comenzar una lista (solo se aplica a partir de dicha lista en adelante):

`\renewcommand \labelitemi{ESTILO}`

En la definición de estos estilos podemos especificar, en vez de “labelitemi”:

- *labelitemii* para listas no numeradas de Nivel 2
- *labelitemiii* para listas no numeradas de Nivel 3
- *labelitemiv* para listas no numeradas de Nivel 4

Entre los estilos que podemos especificar están:

- Cualquier texto o símbolo que deseemos, con el estilo que deseemos (negrita, cursiva, etc.).
- `\square`: □
- `\bullet`: ●

- `\ast`: *
- `\cdot`: ·
- `\blacksquare`: ■

1.13.2. Estilos en listas numeradas

Podemos personalizar los estilos de las listas numeradas mediante mediante el siguiente código en el preámbulo (se cambiaría el estilo de todas las listas de todo el documento) o justo antes de comenzar una lista (solo se aplica para dicha lista):

`\renewcommand \labelenumi{ESTILO}`

En la definición de estos estilos podemos especificar, en vez de “labelenumi”:

- *labelenumii* para listas numeradas de Nivel 2
- *labelenumiii* para listas numeradas de Nivel 3
- *labelenumiv* para listas numeradas de Nivel 4

Entre los estilos que podemos especificar están:

- I. `\arabic{enumi}` - Estilo de números (estilo por defecto) (1, 2, 3, etc.).
- II. `\roman{enumi}` - Estilo de números romanos (i, ii, iii, etc.). Poner *enumi*, *enumii*, *enumiii* ó *enumiv* según el nivel de la lista.
- III. `\Roman{enumi}` - Estilo de números romanos en mayúsculas (I, II, III, IV, etc.).
- IV. `\alph{enumi}` - Estilo de letras (a, b, c, etc.).
- V. `\Alph{enumi}` - Estilo de letras mayúsculas (A, B, C, etc.).

También podemos especificar el número inicial del contador en nuestras listas numeradas. Para ello usaremos el comando `\setcounter{enumi}{VALOR INICIAL}` (pondremos *enumi*, *enumii*, etc. según deseemos) dentro de la definición de nuestra lista:

5. Primero

6. Segundo

7. Tercero

1.14. División en múltiples archivos

Una de las grandes dificultades que nos encontramos cuando comenzamos a trabajar con \LaTeX es lo dificultoso que es navegar por el código de un documento extenso. Para evitar dicho problema, aunque es opcional, \LaTeX permite dividir un documento en diversos ficheros usando el comando `\include{RUTA ARCHIVO SIN EXTENSION .TEX}`. De esta forma, podemos tener ficheros para cada capítulo, sección, etc. La división en ficheros la podemos realizar como deseemos.

En este documento de ejemplo, los siguientes capítulos se encuentran en ficheros aparte, dentro de la carpeta *capitulos*: *cap02.tex* y *cap03.tex*.

Como se puede observar, el documento final incluirá todo el texto y comandos tanto del fichero principal como de los ficheros adicionales.

Capítulo 2

Tablas, figuras, ecuaciones e índices

2.1. Figuras

Para poder incluir figuras en nuestros documentos es necesario incluir el paquete “graphicx”. Además, es conveniente (aunque opcional) meter todas las figuras de nuestro documento en una carpeta dedicada a ello. \LaTeX permite predefinir una carpeta para incluir las figuras de todo nuestro documento usando el siguiente comando en el preámbulo: `\graphicspath{CARPETA }`.

A diferencia de los procesadores de texto habituales, las figuras se *referencian*, no se *insertan* en el documento.

Para referenciar una figura se utiliza en comando `\includegraphics{NOMBRE IMAGEN }` (el nombre de la imagen también puede hacer referencia a una subcarpeta). En el nombre de la imagen no será necesario incluir la extensión. Además, \LaTeX no solo permitirá imágenes PNG, JPEG, etc., sino también gráficos vectoriales en formato PDF, EPS, etc. De hecho, se recomienda usar **imágenes en formato vectorial siempre que sea posible**.

Veamos un ejemplo de imagen vectorial:

L^AT_EX

Las figuras pueden también escalarse, rotarse, etc. según nuestras necesidades usando una variante del comando anterior: `\includegraphics[OPCIONES]{NOMBRE IMAGEN}`.

Al igual que en el resto de comandos de L^AT_EX, las opciones se separan por comas. Las opciones más habituales disponibles para este comando son:

- *width*: Ancho de la imagen (en cm, mm, pt, etc.). Existen dos anchos especiales: `\textwidth` y `\columnwidth`. Con estos anchos podemos hacer la imagen del mismo ancho que el texto o columna actual, o bien aplicar un factor sobre dicho ancho. Por ejemplo `\includegraphics[width=0.5\textwidth]{NOMBRE IMAGEN}` incluirá una imagen con un ancho del 50 % del ancho total del texto.
- *height*: Alto de la imagen.
- *scale*: Factor de escala sobre el tamaño original de la imagen.
- *angle*: Ángulo de rotación de la imagen en grados y en el sentido antihorario.

Veamos un ejemplo:



Se podrán observar como L^AT_EX intentará que las imágenes se autoposicionen: se alterará el interlineado, se ampliarán márgenes, etc. Esto puede suponer un problema en diversos casos (sobre todo cuando tenemos restricciones en el número de páginas). En secciones posteriores aprenderemos como arreglar este problema.

2.2. Tablas

La definición de tablas en \LaTeX es quizás de las cuestiones más complejas que posee.

La manera más básica de definir una tabla es como sigue:

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

Cuya visualización sería la siguiente:

Celda 1	Celda 2	Celda 3
Celda 4	Celda 5	Celda 6
Celda 7	Celda 8	Celda 9

Se puede observar como se deben de especificar las columnas primero, con la alineación de cada una:

- *c*: Columna centrada
- *r*: Columna alineada a la derecha
- *l*: Columna alineada a la izquierda

Las “ \updownarrow ” permiten especificar los bordes en las columnas. Se pueden poner dobles o simples, externos o internos, y allá donde deseemos.

A nivel de contenidos, las diferentes columnas se separan con el símbolo $\&$. Las filas con el símbolo “ \backslash ”, que en \LaTeX permite además crear un salto de línea.

Para dividir las filas, de manera opcional también podemos usar el comando $\backslash\textbf{hline}$, que crea una línea horizontal.

Podemos especificar líneas horizontales entre determinadas columnas únicamente usando el comando $\backslash\textbf{cline}\{\text{COL. INICIAL} - \text{COL. FINAL}\}$.

2.2.1. Tamaños de columnas

Para poder especificar tamaños de columnas fijos es necesario incluir en el preámbulo el paquete “array”.

Podemos crear columnas con un tamaño fijo usando para ello, en la especificación de las columnas, el tipo especial *m* de la siguiente forma:

```
\begin{tabular}{m{TAMAÑO} | m{TAMAÑO} | m{TAMAÑO} }
Celda 1 & Celda 2 & Celda 3 \\
\hline
Celda 4 & Celda 5 & Celda 6 \\
Celda 7 & Celda 8 & Celda 9
\end{tabular}
```

Cuya visualización sería la siguiente:

Celda 1	Celda 2	Celda 3	Celda 4
Celda 5	Celda 6	Celda 7	Celda 8
Celda 9	Celda 10	Celda 11	Celda 12

2.2.2. Combinación de filas y columnas

Para poder combinar filas y columnas es necesario incluir el paquete “multirrow” en el preámbulo.

Un ejemplo de tabla con filas y columnas combinadas es el que sigue¹:

Comunidades autónomas	
Nombre	Provincia
Andalucía	Granada
	Almería
	Jaén
	Málaga
	Cádiz
	Córdoba
	Sevilla
	Huelva
Extremadura	Cáceres
	Badajoz

2.2.3. Importación desde CSV

L^AT_EX permite crear tablas a partir de archivos CSV externos. Para ello tendremos que importar el paquete “csvsimple”. Dicho paquete permite auto-

¹Mirar el código de este documento para entender cómo se especifican

generar las tablas mediante el comando `\csvautotabular{fichero CSV}` o bien usar los datos del CSV para generar nuestra propia tabla personalizada.

Por ejemplo, el comando `\csvautotabular{csv/ejemplo.csv}` generaría la siguiente tabla:

Month	1958	1959	1960
JAN	340	360	417
FEB	318	342	391
MAR	362	406	419
APR	348	396	461
MAY	363	420	472
JUN	435	472	535
JUL	491	548	622
AUG	505	559	606
SEP	404	463	508
OCT	359	407	461
NOV	310	362	390
DEC	337	405	432

Si deseamos tener un control exhaustivo sobre cómo se visualizan los diferentes elementos de la tabla, podemos usar el comando `\csvreader[opciones]{fichero CSV}{columnas}{formato columnas}`. Las *columnas* son una secuencia de “nombre=`\clave`” donde a cada nombre de columna del CSV le asociaremos un comando de \LaTeX personalizado. En *formato columnas* usaremos dichos comandos para aplicarle un formato a los valores de dichas columnas en cada fila. También podremos usar el comando `\thecsvrow` para referirnos al número de la fila.

Un ejemplo de su uso se muestra a continuación². La documentación completa del paquete “csvsimple” se encuentra disponible en <https://tools.ietf.org/doc/texlive-doc/latex/csvsimple/csvsimple.pdf>.

²Véase el código \LaTeX para comprender cómo se ha generado la tabla mostrada

#	Mes	1958	1959	1960
1	JAN	340	360	417
2	FEB	318	342	391
3	MAR	362	406	419
4	APR	348	396	461
5	MAY	363	420	472
6	JUN	435	472	535
7	JUL	491	548	622
8	AUG	505	559	606
9	SEP	404	463	508
10	OCT	359	407	461
11	NOV	310	362	390
12	DEC	337	405	432

2.3. Listados de código

A lo largo de este documento hemos tenido que incluir código de \LaTeX para comprender su sintaxis. Existen dos comandos para hacer listados de código de manera mucho más sencilla a como hemos visto hasta ahora.

El primero de ellos es `\begin{verbatim} ... \end{verbatim}`, que permite incluir texto “literal” en nuestros documentos, sin que \LaTeX interprete ninguno de sus contenidos como un posible comando. El siguiente ejemplo ilustra el uso de este comando³:

```
\begin{tabular}{m{2.5cm}|m{25mm}|m{40pt}|m{0.15\textwidth}}
Celda 1 & Celda 2 & Celda 3 & Celda 4 \\
\hline
Celda 5 & Celda 6 & Celda 7 & Celda 8 \\
Celda 9 & Celda 10 & Celda 11 & Celda 12
\end{tabular}
```

Obsérvese como *verbatim* además usa un tipo de fuente especial para estos fragmentos de texto.

El otro comando, mucho más avanzado que *verbatim*, es `\begin{lstlisting} ... \end{lstlisting}`. Dicho comando requiere incluir el paquete “listings” en el preámbulo del documento. Este comando se diferencia de *verbatim*

³Observar PDF y código

en que permite incluir ficheros externos, realzar partes del código con colores o tipos de letras diferentes, etc. Además, incluye soporte por defecto a multitud de lenguajes de programación, aunque podemos especificar nuestro propio esquema de realzado para cualquier texto o lenguaje. El listado de lenguajes soportados por defecto podemos consultarlo en: https://es.overleaf.com/learn/latex/Code_listing#Reference_guide.

El siguiente ejemplo ilustra el uso de este comando para mostrar un código del lenguaje R⁴:

```
if(num < 0) {
  print("El factorial no existen para numeros negativos")
} else if(num == 0) {
  print("El factorial de 0 es 1")
} else {
  for(i in 1:num) {
    factorial = factorial * i
  }
  print(paste("El factorial de", num, "es", factorial))
}
```

El ejemplo que se muestra a continuación incluye un código R desde un fichero externo (carpeta “codigo_externo/ejemplo.R”):

```
if(num < 0) {
  print("El factorial no existen para numeros negativos")
} else if(num == 0) {
  print("El factorial de 0 es 1")
} else {
  for(i in 1:num) {
    factorial = factorial * i
  }
  print(paste("El factorial de", num, "es", factorial))
}
```

Entre las opciones de este comando de importación de código externo están *firstline* y *lastline*, que permiten especificar el rango de líneas de código a mostrar en el documento.

⁴Observar PDF y código

2.4. Ecuaciones

El soporte a ecuaciones en L^AT_EXes ampliamente conocido. De hecho, su funcionalidad es tan amplia y avanzada que hay personas que piensan que L^AT_EXes un lenguaje exclusivamente para “matemáticos” (¡nada más lejos de la realidad!).

Para crear una ecuación en línea simplemente deberemos escribir dicha ecuación entre `$`. Por ejemplo el código `$x^2+x+1=y_1$` generaría como resultado⁵ $x^2 + x + 1 = y_1$.

Si las ecuaciones se escriben entre `$$`, entonces L^AT_EXreservará una línea específica para dichas ecuaciones y unos márgenes superiores e inferiores.

Veamos algunos ejemplos para tener una referencia de la escritura básica de ecuaciones (dicha escritura esta fuera del ámbito de esta guía):

$$[x^n + y^n = z^n]$$

$$(x_1 + x_2 = 4)$$

$$E = mc^2$$

$$2 \times 2 = 4$$

$$2 \div 2 = 4$$

$$\frac{1}{2} = 0,5$$

$$\sqrt{\pi}$$

$$x \in \Re$$

$$\prod_{i=a}^b f(i)$$

$$\sum_{n=1}^{\infty} 2^{-n} = 1$$

$$\lim_{x \rightarrow \infty} f(x)$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

⁵véase código y PDF a la vez

$$\int_0^1 f(x) dx$$

$$\oint_V f(s) ds$$

$$\begin{pmatrix} 1 & 2 & 3 \\ a & b & c \end{pmatrix}$$

Es importante notar que L^AT_EX tiene soporte para todas las letras griegas dentro de las ecuaciones, usando para ello “\nombre letra”. Por ejemplo, “\beta” generaría la letra griega β . Veamos algunos ejemplos⁶:

$$\alpha_1^n + \beta^n = \rho_{n,1}$$

$$\int_{\epsilon}^{\gamma} x^n dx = \frac{\gamma^{n+1}}{n+1} - \frac{\epsilon^{n+1}}{n+1} + C$$

L^AT_EX además incluye infinidad de comandos adicionales para operadores, símbolos, matrices, etc. Muchos de ellos se pueden consultar en https://es.overleaf.com/learn/latex/List_of_Greek_letters_and_math_symbols.

2.5. Posicionamiento

Como hemos observado previamente, L^AT_EX es capaz de posicionar automáticamente nuestras figuras o tablas. Sin embargo, en muchas ocasiones es necesario realizar dichos posicionamientos manualmente. En las siguientes subsecciones aprenderemos cómo hacer esto.

Para posicionar nuestras figuras deberemos incluirlas dentro entre los comandos `\begin{figure}[OPCIONES POSICION] ... \end{figure}`.

Por ejemplo, podríamos usar el siguiente código para posicionar manualmente una de las imágenes que hemos usado previamente en nuestros ejemplos:

```
\begin{figure}[h]
  \includegraphics{LaTeX_vectorial}
\end{figure}
```

Cuyo resultado sería:

Entre las opciones de posicionamiento se encuentran las siguientes:

⁶Mirar detenidamente el código y el resultado en el PDF

L^AT_EX

- *h*: Posicionar “aquí”, posiciona, si se puede, en el mismo lugar del texto donde ponemos nuestro código de inserción de la figura.
- *t*: Posicionar en la parte superior de la página actual.
- *b*: Posicionar en la parte inferior de la página actual.
- *p*: Posicionar en una página especial al final de todo el documento, junto al resto de figuras que posicionemos ahí.

Podemos combinar las opciones previas. Por ejemplo podemos poner como opción *hb*, para indicar que se posicione la figura exáctamente donde hemos puesto el código de inserción, pero si no se pudiese, que entonces se posicione en la parte inferior de la página.

También podemos agregar el símbolo “!” al final de las opciones de posicionamiento, para indicar que L^AT_EXevite hacer cualquier posicionamiento automático en ningún caso. Por ejemplo, la opción “h!” pondría en cualquier caso la figura justo en el lugar donde hemos escrito el código de inserción de la misma.

Para posicionar nuestras tablas, será necesario crearlas entre los comandos `\begin{table}[OPCIONES POSICION] ... \end{table}`. Por lo demás, es equivalente al posicionamiento de figuras.

Adicionalmente, podemos usar `\begin{wrapfigure} ... \end{wrapfigure}` ó `\begin{wraptable} ... \end{wraptable}`, para crear figuras o tablas, respectivamente, que se posicionen de tal manera que el texto del documento las envuelva. Estos comandos estarán disponibles una vez incluyamos el paquete “wrapfig” en el preámbulo del documento.

Por ejemplo, si usamos el siguiente código:

```
Praesent in sapien. Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Duis fringilla tristique neque. Sed interdum
```

libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin.

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

Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin.

Obtendremos como resultado:

Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin.

Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin.

Celda 1	Celda 2	Celda 3	Celda 4
Celda 5	Celda 6	Celda 7	Celda 8

Nótese como a continuación de `\begin{wrapXXX}` hay que especificar el posicionamiento y el tamaño de la figura o tabla, siendo las opciones de posicionamiento:

- *r*: derecha
- *l*: izquierda
- *c*: centro
- *i*: interno
- *o*: externo

2.6. Leyendas

2.6.1. Figuras o tablas

Para poner una leyenda a nuestra figura o tabla, simplemente se utilizará el comando `\caption{LEYENDA}` entre los comandos `\begin{figure} ... \end{figure}`, `\begin{table} ... \end{table}`, `\begin{wrapfigure} ... \end{wrapfigure}` ó `\begin{wraptable} ... \end{wraptable}`.

L^AT_EX se encargará de numerar adecuadamente las leyendas, aplicarles un estilo, posicionarlas, etc.

Por ejemplo, el código:

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.3\textwidth]{LaTeX_vectorial}
  \caption{Leyenda de una figura}
\end{figure}
```

```
\begin{figure}[h]
  \caption{Leyenda de otra figura (arriba)}
  \centering
  \includegraphics[width=0.3\textwidth]{LaTeX_vectorial}
\end{figure}
```

Generará como resultado:



Figura 2.1: Leyenda de una figura (debajo)

Figura 2.2: Leyenda de otra figura (encima)



Nótese como se ha utilizado el comando `\centering`, muy habitual al especificar leyendas, para centrar la figura.

2.6.2. Subfiguras

Entre los comandos `\begin{figure}[OPCIONES POSICION] ... \end{figure}` podemos especificar más de una figura a modo de “subfiguras”, cada una con su propia leyenda. Para ello, primeramente deberemos incluir el paquete “subcaption” dentro del preámbulo.

Por ejemplo, el código:

```
\begin{figure}[h]
  \begin{subfigure}{0.5\textwidth}
    \includegraphics[width=0.9\linewidth]{LaTeX_vectorial}
    \caption{Leyenda 1}
  \end{subfigure}
  \begin{subfigure}{0.5\textwidth}
    \includegraphics[width=0.9\linewidth]{LaTeX_vectorial}
    \caption{Leyenda 2}
  \end{subfigure}

  \begin{subfigure}{\textwidth}
    \centering
    \includegraphics[width=0.8\linewidth]{LaTeX_vectorial}
    \caption{Leyenda 3}
  \end{subfigure}

  \caption{Leyenda para las tres figuras}
\end{figure}
```

Generaría el siguiente resultado:

2.6.3. Código

Para incluir una leyenda en un código especificado entre `\begin{lstlisting} ... \end{lstlisting}` ó `\lstinputlisting` usaremos la opción “caption” de dichos comandos.

Por ejemplo el comando `\lstinputlisting[language=R, caption=Ejemplo]{codigo_externo/ejemplo.R}` generará:

Listing 2.1: Código R de ejemplo

```
if(num < 0) {
  print("El factorial no existen para numeros negativos")
}
```



(a) Leyenda 1

(b) Leyenda 2



(c) Leyenda 3

Figura 2.3: Leyenda para las tres figuras

```

} else if(num == 0) {
    print("El factorial de 0 es 1")
} else {
    for(i in 1:num) {
        factorial = factorial * i
    }
    print(paste("El factorial de", num, "es", factorial))
}

```

2.7. Etiquetas y referencias cruzadas

Después de escribir cualquier comando que genere una numeración (capítulos, secciones, leyendas, etc.) podremos escribir el comando especial `\label{nombre etiqueta}`. Dichas etiquetas nos permitirán referenciar los números generados por \LaTeX mediante el comando complementario `\ref{nombre etiqueta}`. De esta forma, si cambia el documento, se reestructura, etc., jamás tendremos que preocuparnos de revisar las numeraciones de las referencias cruzadas.

Por ejemplo, a continuación vamos a referenciar la Figura 2.4 y la Sección 2.8. En el primer caso, hemos puesto la etiqueta “fig:refs” tras la leyenda de la figura. En el segundo caso hemos puesto la etiqueta “sec:indices” tras el comando para generar la sección “Índices” (2.8).



Figura 2.4: Figura de ejemplo

Para etiquetar ecuaciones será necesario usar los comandos `\begin{equation}` ... `\end{equation}` para escribir nuestras ecuaciones en vez de `$` ó `$$`. Dentro podremos incluir el comando `\label`.

2.8. Índices

Existen principalmente tres comandos para generar índices:

- `\tableofcontents`: Índice de capítulos, secciones, etc.
- `\listoffigures`: Índice de figuras.
- `\listoftables`: Índice de tablas.

Como ejemplo, al final de este capítulo se han incluido un índice de figuras.

Estos comandos tienen multitud de opciones adicionales que pueden consultarse en https://es.overleaf.com/learn/latex/Table_of_contents, <https://es.overleaf.com/learn/latex/Indices> y https://es.overleaf.com/learn/latex/Lists_of_tables_and_figures.

Por otro lado, podemos agregar nuestros propias entradas de índices personalizadas con el comando `\addcontentsline{toc}{tipo entrada}{título entrada}`. El tipo de entrada corresponde con *chapter*, *section*, *subsection*, etc., y permitirá que L^AT_EX formatee correctamente la entrada agregada.

Índice de figuras

2.1. Leyenda de una figura (debajo)	22
2.2. Leyenda de otra figura (encima)	22
2.3. Leyenda para las tres figuras	24
2.4. Figura de ejemplo	25

Capítulo 3

Gestión bibliográfica

L^AT_EX permite crear una bibliografía, y gestionar las citas a las diferentes entradas bibliográficas. Existen dos maneras de crear dichas sección: manualmente o a partir de una base de datos bibliográfica externa.

3.1. Entradas manuales y citas

Para crear una sección de bibliografía manual deberemos incluir diversas entradas entre los comandos `\begin{thebibliography}{nº entradas}` ... `\end{thebibliography}`.

Las entradas se crean mediante el comando `\bibitem{etiqueta}` *Entrada bibliográfica completa (autores, título, etc.)*.

Por otro lado, para citar en nuestros documentos cualquier entrada se utilizará el comando `\cite{etiqueta/s}`. Las numeraciones de entradas, forma de citar, etc. lo gestiona L^AT_EX automáticamente. Sin embargo, el estilo de las entradas bibliográficas hay que gestionarlo manualmente (orden de autores, formato de títulos, etc.).

3.2. Bases de datos bibliográficas externas y estilos automáticos

Podemos incluir una base de datos bibliográfica externa que siga el formato Bibtex (muy común en software de gestión bibliográfica y soportado

por las principales editoriales). El comando para incluir una base de datos bibliográfica externa es `\bibliography{nombre archivo Bibtex}`.

La principal ventaja, además de que podemos tener nuestras propias bases de datos compartidas entre diferentes documentos, es que el estilo de las entradas bibliográficas lo gestionará automáticamente \LaTeX . Para definir el estilo de las entradas bibliográficas se utilizará el comando `\bibliographystyle{estilo}`.

Entre los estilos predeterminados de \LaTeX están¹: *abbrv*, *acm*, *abbrv*, *alpha*, *apalike*, *IEEEtran*, *plain*, *siam*, *unsrt*.

Al final de este documento se ha incluido una sección de bibliografía, extraída del fichero “bibliografia.bib”². Se recomienda observar la bibliografía generada. Si se desea consultar más información, puede acudir a las referencias [1, 2].

Se pueden observar diversas entradas bibliográficas de ejemplo a través de las siguientes citas: [3], [4].

Se podrá observar, si se abre el archivo Bibtex que \LaTeX incluirá únicamente las entradas bibliográficas que se citen en el texto, omitiendo las que no.

¹Hay infinidad más a través de paquetes especializados. También podemos definir nuestros propios estilos.

²El formato de este tipo de bases de datos bibliográficas es ajeno a este documento. Puede usarse software como JabRef, Bibdesk, Mendeley, etc. para generar dichas bases de datos.

Bibliografía

- [1] (2021, May). [Online]. Available: <https://es.overleaf.com/learn>
- [2] M. Goossens, F. Mittelbach, and A. Samarin, *The L^AT_EX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.
- [3] G. Grätzer, *More Math Into LaTeX, 4th Edition*. Springer, 1996.
- [4] L. Lamport, *Latex Document Preparation System Users (2nd edition)*. Pearson, 1994.