

---

---

# OTEA - Curso básico de L<sup>A</sup>T<sub>E</sub>X

---

---

Por  
Pascal



## UNIVERSIDAD COMPLUTENSE MADRID

Grado en Ingeniería Informática  
FACULTAD DE INFORMÁTICA

Oficina de Software Libre y Tecnologías Abiertas  
OTEA - Curso básico de L<sup>A</sup>T<sub>E</sub>X

MADRID, 2018–2019



# Sobre TEF<sub>L</sub>ON

TEFLON(CC0 1.0(DOCUMENTACIÓN) MIT(CÓDIGO))ES UNA PLANTILLA DE L<sup>A</sup>T<sub>E</sub>X CREADA POR DAVID PACIOS IZQUIERDO CON FECHA DE ENERO DE 2018. CON ATRIBUCIONES DE USO CC0.

Esta plantilla fue desarrollada para facilitar la creación de documentación profesional para Trabajos de Fin de Grado o Trabajos de Fin de Máster. La versión usada es la 1.3.

V:1.3 OVERLEAF V2 WITH PDFL<sup>A</sup>T<sub>E</sub>X, MARGIN 1IN, NO-BIB

## Contacto

**Autor:** DAVID PACIOS IZQUIERO

**Correo:** dpacios@ucm.es

**ASCII:** asciifdi@gmail.com

DESPACHO 110 - FACULTAD DE INFORMÁTICA



# Índice general

	Página
<b>1. Capítulo 1</b>	<b>1</b>
1.1. ¿Qué es $\LaTeX$ y por qué hay que aprenderlo?	1
1.2. Ventajas respecto a la ofimática	2
1.3. Ejemplos de documentos perfectos en $\LaTeX$	2
1.4. Herramienta online: Overleaf	10
1.5. Partes de un documento	12
1.6. ¿Qué es un comando?	14
1.7. Primeros pasos con $\LaTeX$	14
1.7.1. Modificadores básicos I	14
1.7.2. Modificadores básicos II	16
1.7.3. Modificadores básicos III	16
1.7.4. Modificadores básicos IV	17
1.7.5. Modificadores básicos V	18
1.7.6. Combinación de modificadores básicos	20
1.7.7. Apartados, subapartados, título, autor y fecha	20
1.7.8. Ejercicios resueltos	22
<b>2. Capítulo 2</b>	<b>29</b>
2.1. Tablas a mano	29
2.2. Tablas con tablesgenerator	31
2.3. Insertar imágenes con elementos flotantes	35
2.4. Texto con color	36
<b>3. Capítulo 3</b>	<b>39</b>
3.1. Poner una página en landscape	39
3.2. Paquete multicols	39
3.2.1. Separación de columnas	40
3.2.2. Columnas des-balanceadas	41
3.2.3. Delimitando con líneas verticales	43
3.3. Matemáticas MMI	44
3.3.1. Modo matemático	44
3.3.2. Tablas de símbolos básicos	46
3.3.3. Fórmulas simples	48
3.3.4. Estructuras matemáticas: límites, sumatorios	50
3.3.5. Estructuras matemáticas: fracciones y raíces	52
3.3.6. Estructuras matemáticas: integrales	55
3.3.7. Estructuras matemáticas: matrices, determinantes	57

3.3.8.	Estructuras matemáticas: textificación . . . . .	59
3.3.9.	Estructuras matemáticas: Sistemas de ecuaciones . . . . .	61
3.4.	Licenciar documentos . . . . .	63
3.5.	Paquete fancyhdr . . . . .	64
3.6.	Paquete mdframed . . . . .	66
<b>4.</b>	<b>Capítulo 4</b>	<b>69</b>
4.1.	Símbolos en los itemizados . . . . .	69
4.2.	Letras en los enumerados . . . . .	72
4.3.	Circles (números entre círculos) . . . . .	73
4.4.	Pie de página bien en un formato bonito con fancyhdr . . . . .	73
4.5.	Matrices y determinantes . . . . .	74
<b>5.</b>	<b>Capítulo 5</b>	<b>77</b>
5.1.	Ajedrez . . . . .	77
5.1.1.	Tablero . . . . .	77
5.1.2.	Las piezas . . . . .	78
5.2.	Fuentes genéricas . . . . .	80
5.3.	Indentación de párrafos . . . . .	82
5.4.	Espacio entre párrafos . . . . .	84
5.5.	Insercción de documentos PDF . . . . .	84
<b>6.</b>	<b>Capítulo 6</b>	<b>87</b>
6.1.	¿Qué es Beamer? . . . . .	87
6.2.	Creación de bloques . . . . .	88
6.2.1.	Cajas con color . . . . .	90
6.3.	Animaciones . . . . .	92
6.3.1.	Generalidad . . . . .	92
6.3.2.	Especificaciones . . . . .	94
6.3.3.	Comando onslide . . . . .	94
6.3.4.	Comando only . . . . .	95
6.4.	Efectos de transparencia . . . . .	97
6.5.	Comando visible . . . . .	98
6.5.1.	Estilos de la presentación . . . . .	100
<b>7.</b>	<b>Capítulo 7</b>	<b>109</b>
7.1.	Diagrama de barras, círculos . . . . .	109
7.1.1.	Diagrama de barras . . . . .	109
7.1.2.	Diagrama de círculos . . . . .	111
7.2.	Esquema con llaves . . . . .	112
7.3.	Diagramas de flujo . . . . .	113
7.4.	Cambiar nombre de los capítulos con renewcommand . . . . .	115
7.5.	Cambiar color de fondo de folio . . . . .	115
<b>8.</b>	<b>Capítulo 8</b>	<b>117</b>
8.1.	El entorno verbatim . . . . .	117
8.2.	Usando listings para resaltar el código . . . . .	117
8.3.	Importando el código . . . . .	118
8.4.	Dando estilo al código . . . . .	119

8.5.	Listado y nombre del código . . . . .	121
8.6.	Poner palabras clave . . . . .	122
<b>9.</b>	<b>Capítulo 9</b>	<b>125</b>
9.1.	Bibliografía . . . . .	125
9.1.1.	Estilos de bibliografías . . . . .	127
9.1.2.	BibTex . . . . .	128
9.2.	Uso de Google Scholar . . . . .	132
<b>10.</b>	<b>Plantilla básica para escritores</b>	<b>135</b>
10.1.	Introducción . . . . .	135
10.2.	La puntuación . . . . .	135
10.2.1.	La coma . . . . .	135
10.3.	El punto y la coma . . . . .	136
10.3.1.	El punto . . . . .	137
10.3.2.	Los dos puntos . . . . .	138
10.3.3.	Las comillas . . . . .	139
10.4.	Los signos de interrogación y exclamación . . . . .	140
10.5.	El guión . . . . .	141
10.5.1.	Los puntos suspensivos . . . . .	141
10.5.2.	La raya y los paréntesis . . . . .	141
10.6.	La raya en los diálogos . . . . .	142
10.7.	Palabras esdrújulas y sobresdrújulas . . . . .	143
10.8.	Otras normas . . . . .	143
10.9.	Los tiempos verbales en la narración . . . . .	144
10.9.1.	La narración en el pasado . . . . .	144
10.9.2.	La narración en presente . . . . .	145
10.9.3.	Salto temporales . . . . .	146
10.10.	Cursiva . . . . .	146
10.11.	Mayúsculas . . . . .	146
10.12.	Laísmo, leísmo y loísmo . . . . .	147
10.12.1.	Laísmo . . . . .	147
10.12.2.	Loísmo . . . . .	147
10.12.3.	Leísmo . . . . .	148
10.13.	Gerundio . . . . .	148
10.14.	Los títulos . . . . .	148
<b>11.</b>	<b>Guión para TFGs</b>	<b>151</b>
11.1.	Introducción . . . . .	151
11.1.1.	Objetivos . . . . .	151
11.1.2.	Metodología de trabajo . . . . .	151
11.2.	Formato . . . . .	153
11.2.1.	Lengua . . . . .	153
11.2.2.	Imágenes . . . . .	153
11.2.3.	Estilo . . . . .	153
11.3.	Estructura . . . . .	153
11.3.1.	Preliminares . . . . .	154
11.3.2.	Título . . . . .	154
11.3.3.	Subtítulo . . . . .	154

---

11.3.4. Autor/a y año de creación de la obra . . . . .	154
11.3.5. Director/a y departamento al que está adscrito el tutor del TFG .	154
11.3.6. Dedicatoria . . . . .	154
11.3.7. Agradecimientos . . . . .	154
11.3.8. Otras partes . . . . .	154
11.3.9. Resumen . . . . .	154
11.3.10. Índices y glosarios . . . . .	155
11.3.11. Índice o sumario . . . . .	155
11.3.12. Índices generales, tablas e imágenes . . . . .	155
11.3.13. Otras . . . . .	155
11.3.14. Cuerpo principal . . . . .	155
11.3.15. Estado del arte . . . . .	156
11.3.16. Análisis del problema . . . . .	157
11.3.17. Diseño de la solución . . . . .	158
11.3.18. Implementación . . . . .	158
11.3.19. Resultados . . . . .	158
11.3.20. Conclusiones . . . . .	159
11.3.21. Bibliografía . . . . .	160
11.3.22. Referencias a pie de página . . . . .	160
11.3.23. Anexos . . . . .	160
11.4. Tratamientos especiales . . . . .	161
11.4.1. Código fuente . . . . .	161
11.4.2. Defensa pública del TFG . . . . .	161
11.4.3. Diapositivas . . . . .	161
11.4.4. Comunicador . . . . .	162
11.4.5. Estructura de los contenidos . . . . .	162
11.4.6. El día de la presentación . . . . .	162
11.4.7. Recomendaciones . . . . .	162
11.4.8. Malas prácticas . . . . .	162
11.5. Manejo de ideas . . . . .	162
11.6. Excepciones predefinidas . . . . .	163
11.6.1. Excepciones definidas por el usuario . . . . .	163
11.6.2. Uso de SQLCODE y SQLERRM . . . . .	164



# Capítulo 1

# Capítulo 1

## 1.1. ¿Qué es $\LaTeX$ y por qué hay que aprenderlo?

$\LaTeX$  es un sistema de preparación de documentos basado en texto plano. Esta basado en el sistema de lo que escribes, es lo que ves. A través de una serie de comandos, vamos dando una serie de órdenes al programa o al procesador para que lo compile y lo transforme en texto.

Es una herramienta útil y sencilla de utilizar si se conoce el lenguaje. En lo práctico, se puede comparar con lenguajes como HTML.

Además, es de licencia de software libre denominada LPLL, que permite la redistribución y la modificación de los documentos. Hay versiones de  $\LaTeX$  para todo tipo de sistemas y arquitecturas.

Por otro lado,  $\LaTeX$  nos permite modificar el contenido del documento sin que este inflencie al resto. Por lo que podremos eliminar capítulos de un libro sin que se nos desconfigure el documento.

Una vez que hemos definido un poco que es  $\LaTeX$ , vamos a ver por qué es importante aprenderlo.

Primero, es importante aprenderlo para poder redactar documentos de calidad, ya que, es una herramienta que diseña documentos de calidad y nos permite estructurarlo de una manera sencilla.

Segundo, podemos utilizar  $\LaTeX$  mediante múltiples herramientas, ya sea, por programación en texto plano en tiempo real tipo Overleaf o ShareLatex, o por programas que programan en texto plano como  $\TeX$ .



Figura 1.1: Logo Overleaf



Figura 1.2: Logo ShareLatex

Y por último, es importante aprenderlo, ya que, es un sistema muy utilizado en artículos científicos, tesis doctorales y trabajos de fin de grado.

## 1.2. Ventajas respecto a la ofimática

	<b>Word</b>	<b>Latex</b>
Elemento estructurados	No hay diferenciación entre contenido y estructura del texto	Diferenciación entre contenido y estructura del texto
Facilidad de uso	Es muy sencillo, no requiere conocimientos básicos	Requiere conocimientos previos del lenguaje
Inserción de imágenes	Fácil inserción, pero modifica la estructura del texto	Fácil inserción, su inserción no modifica la estructura del texto
Inserción de ecuaciones	Muy complicado, sobre todo para ecuaciones extensas	Muy sencillo, mediante distintos modos se pueden insertar distintas ecuaciones
Presentación de diapositivas	Es intuitivo, permite la inserción de videos	Tiene un paquete para diapositivas, la inserción de videos es complicada
Realización de trabajos de fin de grado	Es complicado insertar bibliografía, cualquier cambio en el capítulo puede modificar toda la estructura del texto	Tiene un paquete para la bibliografía, se puede eliminar cualquier capítulo sin modificar la estructura y es sencillo de utilizar

La ofimática, es una herramienta muy utilizada, con ella, la mayoría de alumnos realizan sus trabajos. Es la más utilizada porque es una herramienta que te viene con el sistema operativo. Pero por otro lado, tenemos otra herramienta como es  $\text{\LaTeX}$ , que nos permite realizar esos mismos textos, pero programándolos mediante comandos.

La comparación entre estas dos herramientas no sería muy útil, ya que, cada usuario decide qué herramienta le es mejor utilizar, con la tabla de arriba queremos vislumbrar para que es mejor cada herramienta.

## 1.3. Ejemplos de documentos perfectos en $\text{\LaTeX}$

Como hemos dicho antes, mediante  $\text{\LaTeX}$  podemos crear todo tipo de documentos, desde un artículo científico, un libro o una tesis doctoral. A continuación se van a mostrar ejemplos de documentos perfectos con su código:

## RECETA - Arroz con cosas

David Pacios

### Ingredientes

- 550 g. pollo
- 550 g. conejo
- 400 g. arroz bomba.
- Caldo de ave o agua.
- Sal, pimienta.
- Azafrán molido.
- Tomate rallado.
- Aceite de oliva.
- Judías verdes planas.
- Judías blancas garrofón.
- Romero.

### Preparación en paellera

Se echa el aceite y un poco de sal y cuando se caliente se mete el pollo y el conejo. Freír bien la carne. Cuando lo veamos un poco dorado se echan las verduras. Rehogar las verduras. Salpimentamos.

Echamos el tomate rallado y lo incorporamos con la carne y las verduras. Ahora se incorpora el agua o caldo de ave (triple que arroz). **Cocer 15 minutos.** (Probar punto de sal después de los 15 minutos). Se mete el azafrán.

Se mete el arroz haciendo una cruz con él para poder esparcirlo mejor. **Los primeros 7 minutos a fuego fuerte.** Se añade un par de ramas de romero. Se baja a fuego medio y a los 5 minutos quitar el romero. Los 4 últimos minutos, a fuego lento. Se retira del fuego y se tapa con un paño 5 minutos.

Este documento esta realizado bajo licencia Creative Commons "CC0 1.0 Universal".



Listing 1.1: Código artículo

```
1 %Preamble
2 \documentclass [ 12pt , a4paper ] { article }
3 \usepackage [ spanish ] { babel }
4 \usepackage [ utf8 ] { inputenc }
5 \usepackage { xparse }
6 \usepackage { multirow }
7 \usepackage [ margin=1in ] { geometry }
8 \usepackage { amsmath , amsthm , amssymb }
9 \usepackage { booktabs }
10 \usepackage { mathpazo }
11 \usepackage { gensymb }
12 \usepackage { textcomp }
13 \usepackage { tikz }
14 \usepackage { esvect }
15 \usetikzlibrary { arrows , positioning , fit , shapes , calc }
16 \usepackage { framed }
17 \usepackage { fancyhdr }
18 \usepackage [
19     type={CC} ,
20     modifier={zero} ,
21     version={1.0} ,
22 ] { doclicense }
23 \pagestyle { fancy }
24 \usepackage { chessboard }
25 \storechessboardstyle { 5x5 } { maxfield=e5 }
26
27 \lhead [ x1 ] { RECETA }
28 \rhead [ z1 ] { Carbonara de Selo }
29 \renewcommand { \headrulewidth } { 0.5 pt }
30 \lfoot [ a1 ] { }
31 \renewcommand { \footrulewidth } { 0.5 pt }
32
33 \usetikzlibrary { calc , angles , positioning , intersections , quotes ,
34     decorations.markings }
35 \usepackage { tkz-euclide }
36 \usetkzobj { all }
37 \usepackage { pgfplots }
38 \usepackage { graphicx }
39 \usepackage { subfigure }
40 \usepackage { float }
41 \usepackage { multicol }
42 \pgfplotsset { compat=1.5 }
43 \usepackage { enumerate }
44 %Cuerpo
45 \begin { document }
```

```

45 \title{RECETA - Arroz con cosas}
46 \author{David Pacios}
47 \date{}
48 \maketitle
49 \noindent
50 \begin{mdframed}
51 \textbf{\textcolor{red}{Ingredientes}}
52 \begin{itemize}
53 \item 550 g. pollo
54 \item 550 g. conejo
55 \item 400 g. arroz bomba.
56 \item Caldo de ave o agua.
57 \item Sal, pimienta.
58 \item Azafrán molido.
59 \item Tomate rallado.
60 \item Aceite de oliva.
61 \item Judías verdes planas.
62 \item Judías blancas garrofón.
63 \item Romero.
64 \end{itemize}
65 \end{mdframed}
66 \subsubsection*{Preparación en paellera}
67 \noindent
68 Se echa el aceite y un poco de sal y cuando se caliente se
    mete el pollo y el conejo. Freír bien la carne. Cuando lo
    veamos un poco dorado se echan las verduras. Rehogar las
    verduras. Salpimentamos.\\
69 \noindent
70 Echamos el tomate rallado y lo incorporamos con la carne y las
    verduras. Ahora se incorpora el agua o caldo de ave (triple
    que arroz). \textbf{Cocer 15 minutos.} (Probar punto de sal
    después de los 15 minutos). Se mete el azafrán.\\
71 \noindent
72 Se mete el arroz haciendo una cruz con él para poder
    esparcirlo mejor. \textbf{Los primeros 7 minutos a fuego
    fuerte}. Se añade un par de ramas de romero. Se baja a fuego
    medio y a los 5 minutos quitar el romero. Los 4 últimos
    minutos, a fuego lento. Se retira del fuego y se tapa con un
    paño 5 minutos.

73
74 \vspace*{\fill}
75 \doclicenseThis
76 \end{document}

```

Lo que vemos arriba es la estructura base de un artículo sencillo, esta estructura la podemos utilizar para escribir recetas o realizar ejercicios sencillos.

A continuación, vamos a mostrar la estructura sencilla de un libro.

---

---

## Arquitectura de Computadores

---

---

Por  
**PASCAL**



**UNIVERSIDAD COMPLUTENSE  
MADRID**

Grado en Ingeniería Informática  
FACULTAD DE INFORMÁTICA

**Arquitectura de Computadores**

MADRID, CURSO 2018–2019

Figura 1.4: Ejemplo de un libro

Listing 1.2: Código libro

```

1  \documentclass [ 12pt , a4paper ] { book }
2  \usepackage [ spanish ] { babel }
3  \usepackage [ utf8 ] { inputenc }
4  \usepackage { afterpage }
5  \usepackage { verbatim }
6  \usepackage { eurosym }
7  \usepackage [ normalem ] { ulem }
8  \useunder { \uline } { \ul } {}
9  \usepackage [ table , xcdraw ] { xcolor }
10 \input { data / settings . tex }
11 \newcommand { \pascal } { \LARGE $ { \displaystyle \mathbf { P ! ! ! ^ { } _
    { \scriptstyle A } } ! ! ! ! ! ! ! ! ; \ ; S ! ! _ { \displaystyle C } \ ! A ! ! ! ^
    { { } _ { \scriptstyle L } } } } $ }
12 %%%%%%%%%%%
13 %
14 %      TeFloN V 1.2
15 %
16 %      Plantilla para TFGs
17 %
18 %      by Blai3e Pascal
19 %      (David Pacios)
20 %
21 %%%%%%%%%%%
22
23 \begin { document }
24 \pagenumbering { Roman }
25 \input { data / title . tex }
26 %\input { data / derechos . tex } No es necesario pero se puede
    agregar , quedaría bien si se imprime en cuero por ejemplo
    porque estaría en la primera página útil , si no , queda raro .
27 %\input { data / auto . tex }
28 %\input { data / derechos . tex }
29 %\input { data / greet . tex }
30
31 \input { data / abstractT . tex }
32 \input { data / index . tex }
33 %\input { Chapters / Resumen . tex }
34 %\input { Chapters / Resumenen . tex }
35 %%%%%%%%%%%
36 \parindent = 0em %LA INDENTACION ES SATÁN
37
38 \input { Chapters / Chapter01 . tex }
39 %\input { Chapters / Chapter01en . tex }
40 \input { Chapters / Chapter02 . tex }
41 \input { Chapters / Chapter03 . tex }

```

```

42 \input{Chapters/Chapter04.tex}
43 \input{Chapters/Chapter05.tex}
44 \input{Chapters/Chapter06.tex}
45 %\input{Chapters/Chapter07.tex}
46 %\input{Chapters/Chapter07en.tex}
47 \listoffigures
48 \listoftables
49 \newpage
50 %\addcontentsline{toc}{chapter}{\protect\numberline{8}
    Bibliografía y enlaces de referencia}%0
51 %\bibliographystyle{unsrt}
52 %\bibliography{biblio}
53
54 \thispagestyle{empty}
55 %%%%%%%%%%% COMENTAR ESTAS DOS LÍNEAS PARA QUITAR
    AUTORÍA
56 \vspace*{\fill} %
57 \begin{verbatim}
58 PASCAL
59 Enero 2019
60 Ult. actualización 1 de Enero de 2019
61 \end{verbatim}
62 \LaTeX{} {\verb lic. LPPL} \& powered by {\small{\teflon{}}}\
    textsc{CC-ZERO}
63
64 \doclicenseThis %
65 %%%%%%%%%%%
66 \end{document}

```

Esta es la estructura base de un libro, luego dentro de cada capítulo tendremos distintos comandos para nombrarlos.

Seguidamente, vamos a mostrar lo que sería una estructura de presentación.



# BoardGame APP

Equipo ASCII



EQUIPO ASCII

Proyecto de aplicación

20 de Diciembre, 2018

1 / 16

Figura 1.5: Ejemplo de presentación

Listing 1.3: Código presentación

```

1  \begin{document}
2  \include los Frames con \include
3
4  \include {Frame/Frame1} %\titulo
5  \include {Frame/Frame2}
6  \include {Frame/Frame3}
7  \include {Frame/Frame4}
8  \include {Frame/Frame5}
9  \include {Frame/FrameHola}
10 \include {Frame/Frame6}
11 \include {Frame/Frame7}
12
13 \include {Frame/Frame8}
14 \include {Frame/Frame9}
15 \include {Frame/Frame10}
16 \include {Frame/Frame11}
17 \include {Frame/Frame12}
18 \include {Frame/Frame13}
19 \include {Frame/Frame14}

```

```
20 \include{Frame/Frame27}  
21 \end{document}
```

## 1.4. Herramienta online: Overleaf

Como hemos nombrado anteriormente, Overleaf es una herramienta que nos permite programar en texto plano en tiempo real.

Para poder utilizarla primero, debemos registrarnos en la página web de la siguiente dirección: <https://www.overleaf.com>

Una vez que hayamos entrado en la dirección veremos lo siguiente:

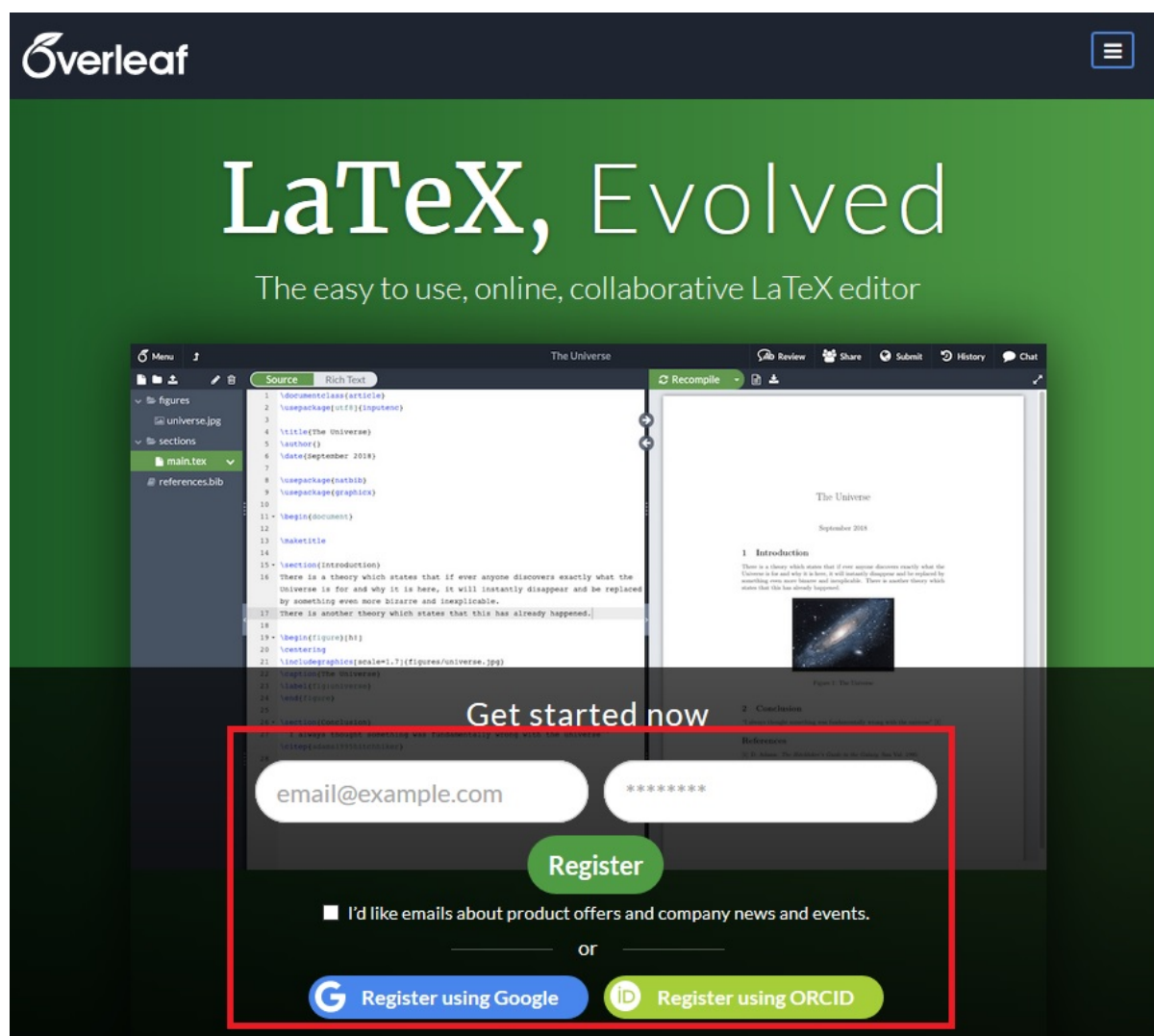


Figura 1.6: Página inicio Overleaf

Como podemos ver, tenemos varias opciones para registrarnos, podemos utilizar nuestra cuenta de Google o la misma cuenta de la ucm.

Seguidamente, una vez estamos registrados, veremos lo siguiente:

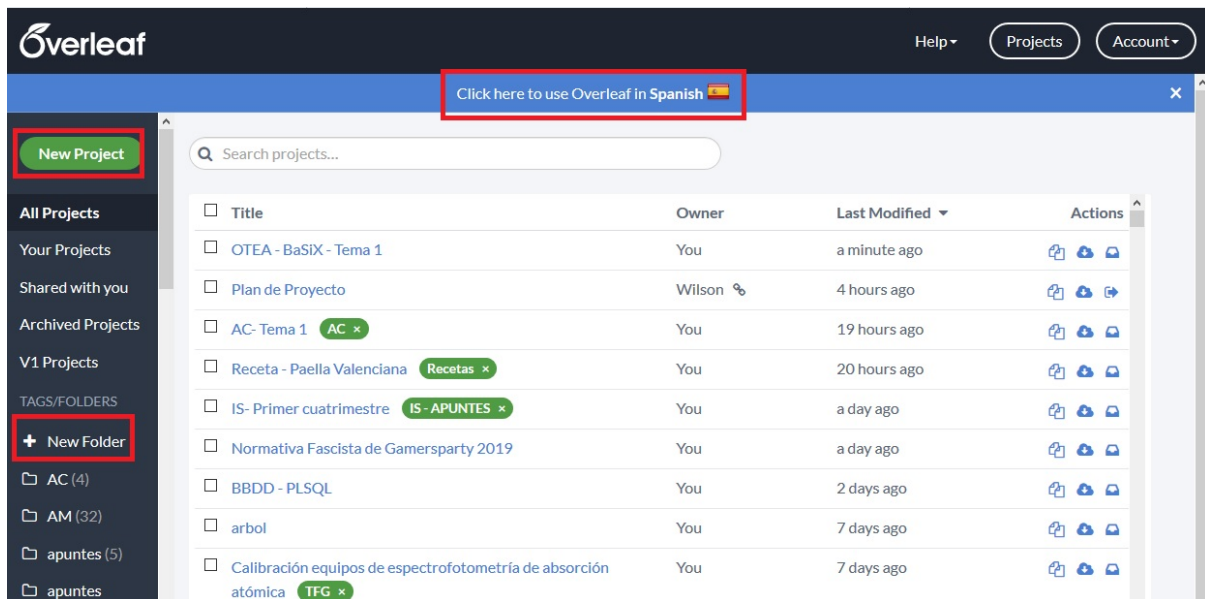


Figura 1.7: Menú Overleaf

Como podemos ver, tenemos la opción de cambiar el idioma en Overleaf, podemos tenerlo en inglés o en español, y si queremos cambiar al castellano sólo tenemos que hacer clic a esa opción. Posteriormente, tenemos la opción de crear una carpeta donde guardar nuestros documentos y también la opción de crear documentos con el botón verde en el que pone New project. Una vez que le demos veremos lo siguiente:

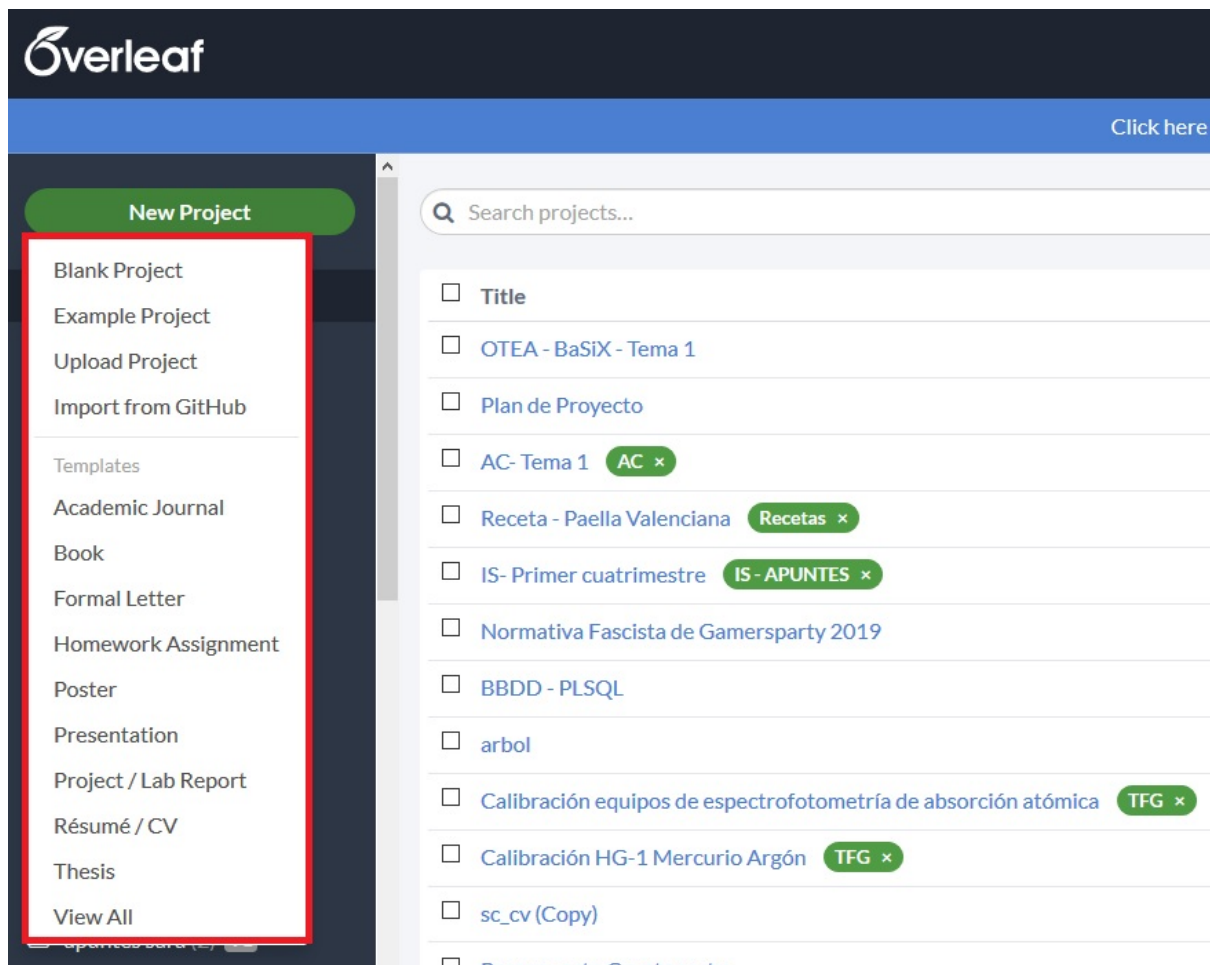


Figura 1.8: Crear nuevo proyecto

Una vez le demos a este botón, nos aparecerá la opción de crear un documento en blanco o elegir entre las plantillas. Y dependiendo de lo que quiera el usuario, podrá elegir entre las distintas opciones. Para los que están aprendiendo todavía, se recomienda utilizar plantillas para empezar a relacionar el comando con su acción.

## 1.5. Partes de un documento

El documento de  $\text{\LaTeX}$  está estructurado por un preámbulo y un cuerpo. Y seguidamente, vamos a ver cuál es el preámbulo y cuál es el cuerpo del texto.

**Preámbulo:**

- Contiene el tipo de documento, tipo de letra, márgenes, espacio entre líneas, etc.
- Le indicamos lo que debe cargar  $\text{\LaTeX}$ .
- Primero se empieza con el tipo del documento.

Listing 1.4: Código preámbulo

```

1 \documentclass {tipo de documento}
2 \usepackage {} Instalación de paquetes

```

En el tipo del documento podemos seleccionar estos tipos:

- **Article:** Para artículos de textos científicos, reportajes...
- **Report:** Para pequeños libros, artículos y tesis que contengan muchos capítulos.
- **Book:** Para libros con una gran cantidad de capítulos.
- **Letter:** Para cartas.
- **Beamer:** Para presentación con diapositivas.

Además en el instalador de paquetes podemos colocar:

- **Préambulo principal:** Indicamos la fuente principal del texto.
- **Idioma:** Con el paquete babel.
- **Imágenes:** Con el paquete graphics.
- **Párrafos:** Con el paquete lipsum.
- **Matrices:** Con el paquete array.
- **Hipervínculos a enlaces:** Con el paquete hyperref o url.
- **Características de la página:** Con el paquete geometry.
- **Situación del texto:** Con el paquete float.
- **Múltiples párrafos:** Con el paquete multicol.
- **Enumeraciones:** Con el paquete enumerate.

**Cuerpo:**

- Contiene lo que le da formato al texto.
- Contiene el entorno del documento, necesario para la compilación.

Listing 1.5: Código preámbulo

```
1 \begin{document}
2 %Comentario en Latex
3 \end{document}
```

Es muy importante cuidar que todos los entornos comiencen con **begin** y terminen con **end**.

## 1.6. ¿Qué es un comando?

Un comando es un carácter especial que determina el comportamiento de nuestro texto en  $\text{\LaTeX}$ .

El comando está dividido en partes. La primera parte del comando está formada por un  $\backslash$  que determina el inicio del comando y posteriormente, la palabra que lo forma es la que da forma al comportamiento de  $\text{\LaTeX}$ .

**Comportamiento en negrita.**

Texto centrado

Listing 1.6: Código comandos

```
1 \textbf{Comportamiento en negrita}.\
2 \begin{center}
3     Texto centrado
4 \end{center}
```

Una evolución de los comandos son los entornos, que comienzan con **begin** y terminan con **end**. Gracias a estos comandos podemos crear los entornos para itemizar, gráficas y muchos otros.

Texto en máquina

- Texto enumerado.
- Texto enumerado.

Listing 1.7: Código entornos

```
1 \texttt{Texto en máquina}
2 \begin{itemize}
3     \item Texto enumerado.
4     \item Texto enumerado.
5 \end{itemize}
```

## 1.7. Primeros pasos con $\text{\LaTeX}$

### 1.7.1. Modificadores básicos I

Hemos visto en apartados anteriores los conceptos de comando y cómo estos modifican el texto. Los primeros que van a ver son los siguientes:

- Negrita.
- Redonda.
- Itálica.

- Máquina de escribir.
- Versalita.

**Negrita:**

Para poner la letra en negrita vamos a utilizar el comando `\textbf{palabra}`. Entre corchetes vamos a colocar la palabra o las palabras que queremos poner en negrita. Y lo vamos a ver de la siguiente forma con su código:

**Palabra** en negrita, **dos palabras** en negrita.

Listing 1.8: Código en negrita

```
1 \textbf{Palabra} en negrita , \textbf{dos palabras} en negrita .
```

**Redonda:**

Para poner la letra en redonda vamos a utilizar el comando `\textrm{palabra}`. Entre corchetes colocaremos la palabra o las palabras que queremos en letra redonda.

Redonda palabra, Texto redondeado.

Listing 1.9: Código en redonda

```
1 \textrm{Redonda} palabra , \textrm{Texto redondeado} .
```

**Itálica:**

Para poner la letra en itálica vamos a utilizar el comando `\textit{palabra}`. Y al igual que en los comandos anteriores, colocaremos la palabra o las palabras que queramos en itálica. Como las siguientes:

*Itálica, Palabras en itálica.*

Listing 1.10: Código en itálica

```
1 \textit{Itálica} , \textit{Palabras en itálica} .
```

**Máquina de escribir:**

Para poner ya sea una palabra o una letra en máquina de escribir, vamos a utilizar el comando `\texttt{palabra}`. Y al igual que en las anteriores entre corchetes vamos a colocar la palabra o las palabras que queremos poner en itálica. Como en las siguientes: Palabra en máquina, Dos palabras en máquina.

Listing 1.11: Código en máquina de escribir

```
1 \texttt{Palabra} en máquina , \texttt{Dos palabras} en máquina .
```

**Versalita:**

Para poner una palabra o una letra en versalita vamos a utilizar el comando `\textsc{palabra}`. Y al igual que en las anteriores, colocaremos entre corchetes las palabras. Para verlo más claro vamos a poner un ejemplo:

VERSALITA una, DOS VERSALITAS en versalita.

Listing 1.12: Código en máquina de versalita

```
1 \textsc{Versalita} una , \textsc{Dos versalitas} en versalita .
```

## 1.7.2. Modificadores básicos II

Una vez hemos visto cómo modificar el texto, vamos a ver cómo podemos mover el texto. Vamos a colocar el texto de la siguiente forma:

- Texto centrado.
- Texto a la derecha o a la izquierda.

### Texto centrado:

Para el texto centrado vamos a utilizar el entorno centrado que comienza con `\begin{center}` y terminan con `\end{center}`. Y como en el apartado anterior vamos a poner un ejemplo:

Ejemplo de texto centrado

Listing 1.13: Código de texto centrado

```
1 \begin{center}
2     Ejemplo de texto centrado
3 \end{center}
```

### Texto a la derecha o a la izquierda:

En este caso, vamos a utilizar un entorno que empezaría con el comando `\begin{flushleft}`, si queremos colocar el texto a la izquierda o empezariamos con el comando `\begin{flushright}` si queremos el texto a la derecha, cerrando cada uno con su `end` respectivamente. Para verlo más claro vamos a poner un ejemplo con los dos entornos:

Texto a la izquierda.

Texto a la derecha.

Listing 1.14: Código de texto a la izquierda y a la derecha

```
1 \begin{flushleft}
2 Texto a la izquierda.
3 \end{flushleft}
4 \begin{flushright}
5 Texto a la derecha.
6 \end{flushright}
```

## 1.7.3. Modificadores básicos III

Ya hemos visto cómo mover el texto, ahora vamos a ver cómo hacer saltos de línea y cómo realizar saltos de página.

### Saltos de línea:

Para realizar saltos de línea podemos utilizar varios comandos, como el comando `\newline` o el comando `\\`. Y además, para el espaciado vamos a utilizar el comando `\quad`. Para verlo más claro vamos a poner un ejemplo:

Este es un salto estándar.

Y este es un salto sin completar

este es    el espaciado.



Listing 1.15: Código de salto de línea y espaciado

```

1 Este es un salto estándar.\\
2 Y este es un salto sin completar \newline
3 este es \quad el espaciado.
```

### 1.7.4. Modificadores básicos IV

Hemos visto cómo realizar los saltos de línea y el espaciado, por lo que vamos a ver seguidamente los saltos de página.

#### Saltos de página:

Para los saltos de comando podemos utilizar el comando `\newpage` que da el salto de página y el comando `\clearpage` si queremos hacer un salto de página cuando hay un elemento flotante.

Vamos a realizar un salto de página.

1



Con elemento flotante

Figura 1.9: Salto de página

Listing 1.16: Código de salto de página

```

1 \section{Introduction}
2 Vamos a realizar un salto de página.
3 \newpage
4 Con elemento flotante
5 \clearpage
6 \end{document}
```

### 1.7.5. Modificadores básicos V

Hemos visto cómo realizar saltos de página, saltos de línea, y por último, vamos a ver cómo modificar el tamaño de las letras o las palabras. Podemos modificar el tamaño de la siguiente forma:

- Tamaño normal.
- Tamaño diminuto.
- Tamaño muy pequeño.
- Tamaño más pequeño.
- Tamaño pequeño.
- Tamaño grande.
- Tamaño más grande.
- Tamaño muy grande.
- Tamaño enorme.
- Tamaño más enorme.

#### Tamaño normal:

El tamaño normal sólo se puede utilizar en la clase de documento **article** y para ello, se utiliza el comando `\normalize{}`. Para verlo más claro vamos a poner un ejemplo:  
Tamaño normal.

Listing 1.17: Código de tamaño normal

```
1 \normalize{Tamaño} normal.
```

#### Tamaño diminuto:

Para poner la palabra en diminuta vamos a utilizar el comando `\tiny{}`. Para verlo más claro vamos a poner un ejemplo:  
Esta palabra es muy pequeña.

Listing 1.18: Código de tamaño diminuta

```
1 Esta {\tiny{palabra}} es muy pequeña.
```

Es muy importante colocar la palabra entre corchetes para que sólo se coloque una palabra en diminuto.

#### Tamaño muy pequeño:

Para poner la palabra en tamaño muy pequeño vamos a utilizar el comando `\scriptsize{}`. Volvemos a poner un ejemplo:  
Esta palabra es muy pequeña.

Listing 1.19: Código de tamaño muy pequeño

```
1 Esta palabra es {\scriptsize{muy}} pequeña.
```

**Tamaño más pequeño:**

Para poner la palabra más pequeña vamos a utilizar el comando `\footnotesize{}`. Y lo veremos con un ejemplo:

Letra más pequeña.

Listing 1.20: Código de tamaño más pequeño

```
1 Letra más {\footnotesize{pequeña}}.
```

**Tamaño pequeño:**

Para poner la palabra en tamaño pequeño vamos a utilizar el comando `\small{}`. Para tenerlo más claro vamos a poner un ejemplo:

Palabra en pequeña.

Listing 1.21: Código de tamaño pequeño

```
1 {\small{Palabra}} en pequeña.
```

**Tamaño grande:**

Para poner una palabra en grande vamos a utilizar el comando `\large{}`. Con el siguiente ejemplo:

Palabra en grande.

Listing 1.22: Código de tamaño grande

```
1 Palabra en {\large{grande}}.
```

**Tamaño más grande:**

Para poner una palabra en un tamaño aún mayor que el anterior vamos a utilizar el comando `\Large{}`. Y lo vamos a ver con el siguiente ejemplo:

Palabra aún más grande.

Listing 1.23: Código de tamaño más grande

```
1 {\Large{Palabra}} aún más grande.
```

**Tamaño muy grande:**

Podemos seguir aumentando el tamaño de la letra con el comando `\LARGE{}`. Con su siguiente ejemplo:

Palabra aún más Grande.

Listing 1.24: Código de tamaño muy grande

```
1 Palabra aún más {\LARGE{Grande}}.
```

**Tamaño enorme:**

Se puede seguir aumentando el tamaño de la letra con el comando `\huge{}`. Con su siguiente ejemplo:

Palabra más grande aún.

Listing 1.25: Código de tamaño enorme

```
1 Palabra más {\huge{grande}} aún.
```

**Tamaño más enorme:**

Y por último, el tamaño enorme requiere el comando `\Huge{}`. Y con su último ejemplo:  
Palabra gigante.

Listing 1.26: Código de tamaño más enorme

```
1 Palabra {\Huge{gigante}}.
```

### 1.7.6. Combinación de modificadores básicos

Hemos visto en dos apartados que podemos resaltar las palabras poniéndolas en negrita o en gigante, pero además de esto, podemos combinar ambos para poner las palabras en negrita y gigante, para ello vamos a utilizar, por ejemplo el comando `\huge{}` seguido del comando `\textbf{}`, ambas entre corchetes. Es muy importante colocar los corchetes, para entenderlo mejor vamos a poner un ejemplo:

**Palabra** en negrita y en grande.

Listing 1.27: Código de tamaño negrita y gigante

```
1 {\huge{\textbf{Palabra}}} en negrita y en grande.
```

### 1.7.7. Apartados, subapartados, título, autor y fecha

Como hemos visto antes, hay muchas clases de documentos cada uno con sus apartados correspondientes.

Para un libro, por ejemplo, para los capítulos vamos a utilizar el comando `\chapter{}`.

Y para los demás documentos, si queremos poner un apartado vamos a utilizar el comando `\section{}` y las siguientes subsecciones requerirán el comando `\subsection{}`, y según queramos colocar más subsecciones colocaremos el prefijo `sub`.

Y por último, en el preámbulo podemos dar un título a nuestro documento con el comando `\title{}`, el nombre del autor con el comando `\author{}` y la fecha con el comando `\date{}`.

Una vez realizado, en el cuerpo sólo tenemos que colocar el comando `\maketitle`.

Para ver todo lo anterior más claro vamos a poner un ejemplo:

## Preambulo- Tema 1

David Pacios

January 2019

### **1 Introduction**

Vamos a realizar un salto de página.

#### **1.1 Segundo apartado**

1

Figura 1.10: Apartado, subapartado y título

Listing 1.28: Código de ejemplo

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3
4 \title { Preambulo - Tema 1 }
5 \author { David Pacios }
6 \date { January 2019 }
7
8 \begin { document }
9 \maketitle
10 \section { Introduction }
11 Vamos a realizar un salto de página .
12 \subsection { Segundo apartado }
13 \end { document }
```

### 1.7.8. Ejercicios resueltos

Una vez hemos visto los conceptos básicos, vamos a aplicarlos a ejercicios.

**Ejercicio 1.** Realiza un artículo en español, en el que ponga Hola mundo, con tu nombre, fecha y de título Ejercicio 1.

Listing 1.29: Ejercicio 1

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage [ spanish ] { babel }
4 \title { Ejercicio 1 }
5 \author { David Pacios }
6 \date { January 2019 }
7
8 \begin { document }
9 \maketitle
10 Hola mundo
11 \end { document }
```

**Ejercicio 2.** Siguiendo con el mismo ejercicio, cámbiale el título al ejercicio 2 y pon en negrita el Hola del Hola mundo.

Listing 1.30: Ejercicio 2

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage [ spanish ] { babel }
4 \title { Ejercicio 2 }
```

```
5 \author{David Pacios}
6 \date{January 2019}
7 \begin{document}
8 \maketitle
9 \textbf{Hola} mundo
10 \end{document}
```

**Ejercicio 3.** Añádele al mismo artículo una sección con el nombre de Sección, y a una subsección con el nombre de Subsección.

Listing 1.31: Ejercicio 3

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage [ spanish ] { babel }
4 \title { Ejercicio 2 }
5 \author { David Pacios }
6 \date { January 2019 }
7
8 \begin { document }
9 \maketitle
10 \section { Sección }
11 \subsection { Subsección }
12 \end { document }
```

**Ejercicio 4.** Realiza otro artículo aparte con las mismas secciones y subsecciones, con un texto centrado en la sección y con un texto a la derecha en la subsección.

Listing 1.32: Ejercicio 4

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage [ spanish ] { babel }
4 \title { Ejercicio 4 }
5 \author { David Pacios }
6 \date { January 2019 }
7
8 \begin { document }
9 \maketitle
10 \section { Sección }
11 \begin { center }
12     Texto centrado
13 \end { center }
14 \subsection { Subsección }
15 \begin { flushright }
```

```
16 Texto a la derecha
17 \end{flushright}
18 \end{document}
```

**Ejercicio 5.** Crea un libro con un capítulo, que ponga capítulo 1 y que contenga una palabra en grande, y otro capítulo que ponga capítulo 2 que contenga una letra en máquina de escribir.

Listing 1.33: Ejercicio 5

```
1 \documentclass {book}
2 \usepackage [utf8] {inputenc}
3 \usepackage [spanish] {babel}
4 \title {Ejercicio 5}
5 \author {David Pacios}
6 \date {January 2019}
7
8 \begin {document}
9 \maketitle
10 \chapter {Capítulo 1}
11 \noindent
12 Esta palabra es {\huge {grande}}.
13 \chapter {Capítulo 2}
14 \noindent
15 Esto está a \texttt {máquina}.
16 \end {document}
```

**Ejercicio 6.** Sigue con el mismo patrón del ejercicio 4, pero realiza un salto de página entre dos secciones pero en un artículo, dividido entre dos secciones. Y además pon centrada la frase en grande, y a la izquierda la frase en máquina.

Listing 1.34: Ejercicio 6

```
1 \documentclass {article}
2 \usepackage [utf8] {inputenc}
3 \usepackage [spanish] {babel}
4 \title {Ejercicio 5}
5 \author {David Pacios}
6 \date {January 2019}
7
8 \begin {document}
9 \maketitle
10 \section {Capítulo 1}
11 \begin {center}
12 Patrón {\huge {grande}}
```



```

13 \end{center}
14 \newpage
15 \section{Capítulo 2}
16 \begin{flushleft}
17 Patrón a la \texttt{izquierda}
18 \end{flushleft}
19 \end{document}

```

**Ejercicio 7.** ¿Qué hay de error en el código? Di porque está mal y qué pondrías para solucionarlo.

```

1 \documentclass{article}
2 \usepackage[utf8]{inputenc}
3 \usepackage[spanish]{babel}
4 \title{Ejercicio 5}
5 \author{David Pacios}
6 \date{January 2019}
7
8 \begin{document}
9 \maketitle
10 \chapter{Capítulo 1}
11 \newpage
12 \chapter{Capítulo 2}
13 \end{document}
14

```

Figura 1.11: Error código

#### Solución ejercicio 7:

Hay que cambiar la clase de documento de `article` a `book`, o cambiar los `\chapter{}` por `\section{}`.

#### Importante!!

Estos ejercicios no sirven si no te lo pasas bien. Escribe aquello que quieras, por muy raro que sea. Se aprende mejor cuando tienes humor.

**Ejercicio 1.** Pon una frase de Hola mundo en un documento que sea un artículo.

**Ejercicio 2.** Pon el siguiente texto en un artículo con las siguientes características:

- Título: Un texto cualquiera.
- Subtítulo: De un tema cualquiera.

Con el siguiente contenido:

- Texto después del título: Vamos a describir las características de un texto sencillo.
- Texto después del subtítulo: El texto no parece muy complicado en un principio.

**Ejercicio 3.** Andrés, el listo del poblado de Calarruras, ha decidido cambiar todos sus pollos por judías mágicas. Pero el vendedor ha pedido antes de nada un listado (con número) del nombre de cada pollo, además de un apartado con una carta de presentación de Andrés. Diseña un par de secciones, una con la carta de presentación y otra con el listado del nombre de los pollos.

**Ejercicio 4.** Escribir un texto no muy extenso con una lista de ventajas del plátano con respecto a los pomelos.

**Ejercicio 5.** Pedro es un alumno extraño, los párrafos pares los empieza escribiendo por la derecha, los párrafos impares por la izquierda. Escribe un texto como si fueras Pedro.

**Ejercicio 6.** Escribe una carta de: animo, agradecimiento, amenaza... Al profesor de L<sup>A</sup>T<sub>E</sub>X. Recuerda usar palabras **resaltadas** y más **grandes** para enfatizar tu enfado. Méteelo todo en una sección.

**Ejercicio 7.** Realiza un libro sencillo con dos capítulos con un título y subtítulo cada uno. Ahora es buena idea intentar hacer un índice.

**Ejercicio 8\*.** Realiza el siguiente texto:

Escribir textos es muy sencillo en L<sup>A</sup>T<sub>E</sub>X hasta llega alguna palabra que toca poner en negrita o en **máquina de escribir**. O que la palabra sea muy grande o muy **pequeña**. L<sup>A</sup>T<sub>E</sub>X nos permite realizar una gran variedad de textos. Podemos realizar desde un artículo hasta un **Libro**.

También nos permite poner:

El texto centrado.

o

El texto a la derecha

o

El texto a la **izquierda**.

#### Corrección

Se subirá en un **.zip** todos los proyectos juntos de L<sup>A</sup>T<sub>E</sub>X Y los corregiré todos dando feedback. No hay notas, venimos a pasarlo bien y aprender. Se darán los resultados con la corrección.



# Capítulo 2

# Capítulo 2

## 2.1. Tablas a mano

El entorno de las tablas nos permite presentar el texto en columnas y para ello, utilizaremos el comando `\begin{tabbing}` y `\end{tabbing}`. Y dentro de este entorno, encontraremos los siguientes comandos:

- `\=`: Fija la posición de los tabuladores.
- `\>`: Salta al siguiente tabulador.
- `\\`: Fin de línea.
- `\kill`: Si una línea finaliza con este comando no se imprime.
- `\+`: Hace que la línea siguiente empiece en el tabulador siguiente al previsto.
- `\-`: Hace que la línea siguiente empiece en el tabulador anterior al previsto.
- `\<`: Salta al tabulador anterior.
- `\``: Hace que el texto anterior se justifique a derecha.
- `\'`: Hace que el texto que sigue vaya hacia el margen derecho.

Una vez hemos visto este entorno básico, vamos a poner un ejemplo sencillo:

```
Nombre NacionalidadDNI
Luis Francesa 0252502A
Nuria Española 5450255C
Esteban Inglesa 8582444E
Lucia Alemana 1201505F
```

Listing 2.1: Ejemplo tabla básica

```
1 \begin{tabbing}
2 Nombre \= Nacionalidad\= DNI\\
```

```

3 Luis \= Francesa \= 0252502A\\
4 Nuria \= Española \= 5450255C\\
5 Esteban \= Inglesa \= 8582444E\\
6 Lucia \= Alemana \= 1201505F\\
7 \end{tabbing}

```

Pero como podemos ver, esta tabla es muy tosca y no nos permite realizar una tabla elaborada.

Para realizar una tabla elaborada vamos a utilizar el entorno `tabular`, que es mucho más fácil de manejar, pero nos permite diseñar una tabla más clara y distribuye bien el texto en columnas.

El entorno tabular está formado por los siguientes comandos: Para empezar el entorno se empieza con `\begin{tabular}[posición]{columnas}` y cerrándolo con `\end{tabular}`.

El comando `columnas` determina la cantidad de columnas y su posición y en ella podemos poner los siguientes comandos:

- `l`: Alinea a la izquierda.
- `c`: Centrar.
- `r`: Alinea a la derecha.
- `|`: Si se coloca entre dos columnas genera una línea vertical.
- `p{ancho}`: Establece el ancho de una columna.
- `@{texto}`: Inserta el texto entre columnas.
- `@{\hspace{ancho}}`: Inserta espacio entre columnas.
- `*{num}{cols}`: num establece la cantidad de columnas y cols el formato de las columnas.

Con todos estos conceptos claros, vamos a poner un ejemplo de una tabla simple:

	Columna 1	Columna 2
Fila 1	Dato 1	Dato 2
Fila 2	Dato 3	Dato 4

Listing 2.2: Ejemplo tabla simple

```

1 \begin{center}
2 \begin{tabular}{|l|l|l|}
3 \hline
4 & Columna 1 & Columna 2 \\ \hline
5 Fila 1 & Dato 1 & Dato 2 \\ \hline
6 Fila 2 & Dato 3 & Dato 4 \\ \hline
7 \end{tabular}
8 \end{center}

```

Como podemos ver, hemos añadido dos opciones más a la tabla, hemos utilizado el entorno `center` para centrar la tabla, y el comando `\hline` para añadir líneas horizontales a nuestra tabla.

Por último, tenemos un entorno mucho más fácil de manejar y sencillo de utilizar. Ese entorno es el `table`. En el cual podemos elegir las posiciones de la tabla que son:

- `b`: En el fondo de la página.
- `h`: En la misma posición que el código fuente.
- `t`: En el principio de la página.
- `p`: Lo pondrá en una página que contenga solo elementos flotantes.
- `!`: Ignora la mayoría de las restricciones.
- `H`: Lo coloca en la misma posición del código fuente como elemento flotante.

Además, le podremos añadir nombres con el comando `\caption{}` y centrar la tabla con el comando `\centering`.

Cuadro 2.1: Tabla con nombre

	Columna 1	Columna 2
Fila 1	Dato 1	Dato 2
Fila 2	Dato 3	Dato 4

Listing 2.3: Ejemplo tabla simple con nombre

```

1 \begin{table}[H]
2 \caption{Tabla con nombre}
3 \begin{tabular}{|l|l|l|}
4 \hline
5 & Columna 1 & Columna 2 \\ \hline
6 Fila 1 & Dato 1 & Dato 2 \\ \hline
7 Fila 2 & Dato 3 & Dato 4 \\ \hline
8 \end{tabular}
9 \end{table}

```

## 2.2. Tablas con `tablesgenerator`

Además de lo que hemos visto en el apartado de cómo realizar tablas, podemos realizarlas de una manera más sencilla con la herramienta `tablesgenerator`.

Primero, vamos a ver cómo funciona la página web.

The screenshot shows the LaTeX Table Generator interface. At the top, there are navigation tabs: LaTeX Tables (selected), HTML Tables, Text Tables, Markdown Tables, MediaWiki Tables, and Contact. Below this is the title 'LaTeX Table Generator' and social media links for Facebook (4659) and Twitter. A menu bar includes File, Edit, Table, Column, Row, Cell, and Help. A toolbar contains icons for bold, italic, underline, and other table editing functions. The main workspace shows a table with columns A, B, and C, and rows 1 to 7. The table content is as follows:

	A	B	C
1	Item		
2	Animal	Description	Price (\$)
3	Gnat	per gram	13.65
4		each	0.01
5	Gnu	stuffed	92.50
6	Emu	stuffed	33.33
7	Armadillo	frozen	8.99

Below the table is a 'Generate' button and a 'Result' section showing the LaTeX code for the table:

```

1 \begin{table}{}
2 \begin{tabular}{llr}
3 \hline
4 \multicolumn{2}{c}{Item} & \\ \cline{1-2}
5 Animal & Description & Price (\$) \\ \hline
6 Gnat & per gram & 13.65 \\
7 & each & 0.01 \\
8 Gnu & stuffed & 92.50 \\
9 Emu & stuffed & 33.33 \\
10 Armadillo & frozen & 8.99 \\
11 \end{tabular}
12 \end{table}

```

Figura 2.1: Captura página web

Como se puede ver, tenemos dos tres zonas, una zona para editar la tabla, una zona para colocarla y una zona para cambiar sus características. Seguidamente, vamos a explicar cómo poner la tabla y cómo se ve:

	A	B	C	D
1	Columna 1	Columna 2	Columna 3	Columna 4
2	Fila 1	Dato 1	Dato 4	Dato 7
3	Fila 2	Dato 2	Dato 5	Dato 8
4	Fila 3	Dato 3	Dato 6	Dato 9

Figura 2.2: Ejemplo de tabla

Una vez que tengamos hecha la tabla, generaremos la tabla con el botón **GENERATE** y la copiaremos con el botón **COPY TO CLIPBOARD**. Y vamos a ver su situación en el tablesgenerator:



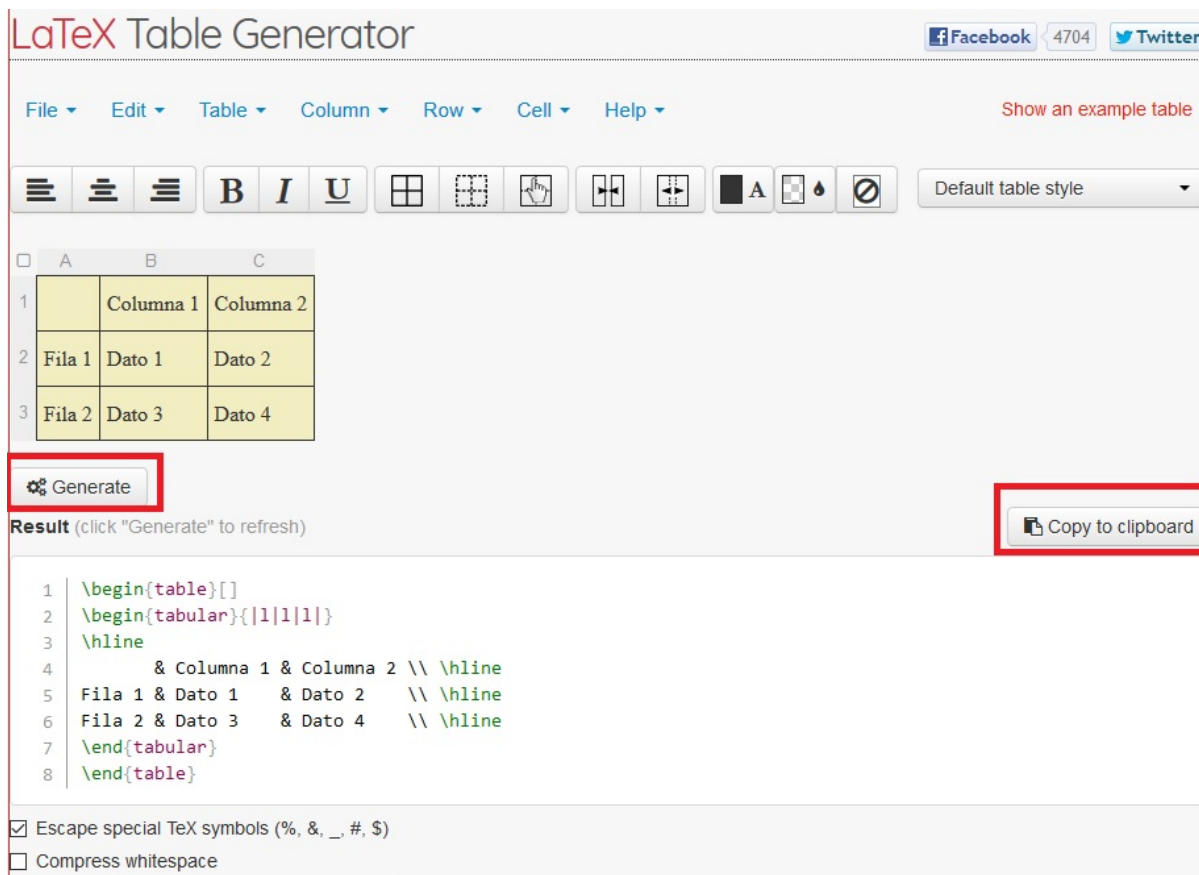


Figura 2.3: Generando tabla y copiándola

Además de lo anterior, podremos centrar o mover a la izquierda o a la derecha las columnas, poner en negrita, cursiva o subrayados, poner bordes, fusionar celdas de filas y de columnas, y dar color a las celdas o a las letras.

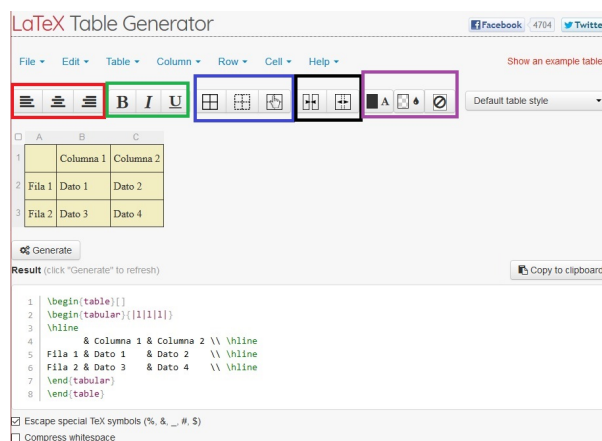


Figura 2.4: Modificación tabla

Y si por alguna casualidad, queremos modificar el tamaño de la tabla, utilizaremos la pestaña **File**:

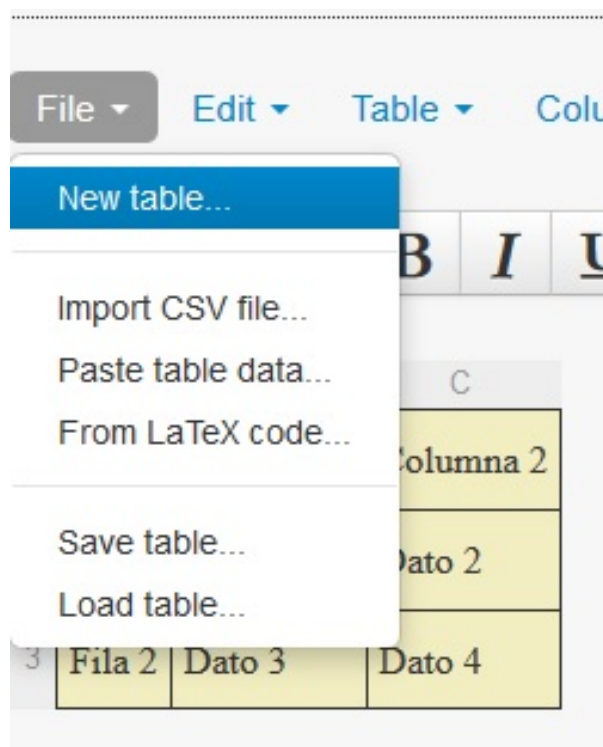


Figura 2.5: Crear una nueva tabla

Y una vez hemos seleccionado **File**, nos aparecerán las siguientes opciones:

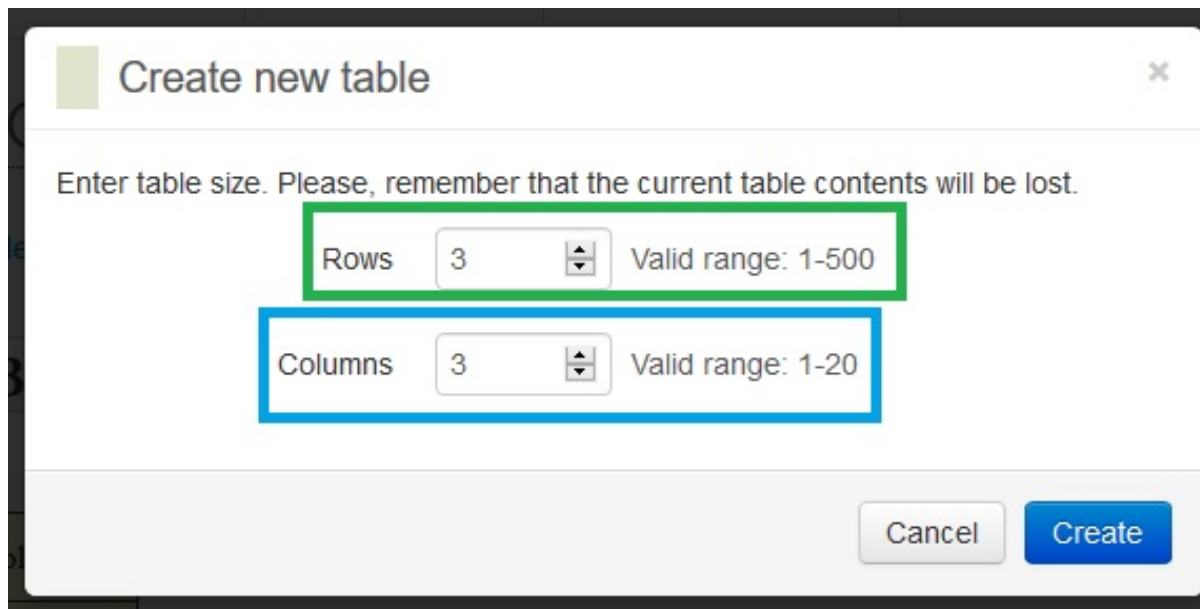


Figura 2.6: Filas y columnas

De aquí hemos señalado tres herramientas importantes:

- **Filas:** En azul están señaladas las filas, con las flechas podemos indicarle que queremos más o menos filas o sino podemos poner el número de filas que queremos directamente.

- **Columnas:** En verde están señaladas las columnas, con las flechas podemos indicarle que queremos más o menos columnas o sino podemos poner el número de columnas que queremos directamente.
- **Create:** En rojo está señalada el botón que tenemos que dar para poder crear nuestra tabla.

Con todo esto, ya podemos crear nuestra tabla en esta página web y pegarla en nuestro documento L<sup>A</sup>T<sub>E</sub>X con mucha facilidad.

## 2.3. Insertar imágenes con elementos flotantes

Lo primero que tenemos que colocar en el preámbulo es el paquete de imágenes que es `\usepackage{graphicx}`.

Después de poner el paquete, escogemos la ruta en la que está la imagen con `\graphicspath{}` e incluir nuestra imagen con el comando `\includegraphics{}`. De momento no vamos a ver cómo se posicionan las imágenes, eso lo veremos en capítulos más adelante. Vamos a ver una imagen colocada de ejemplo:



Figura 2.7: Imagen de ejemplo

Listing 2.4: Código inserción imágenes

```

1 \begin{figure} [H]
2   \centering
3   \includegraphics [width=0.5\textwidth] {Images/TREN.jpg}
4   \caption{Imagen de ejemplo}
5 \end{figure}

```

Después lo que tenemos que hacer es utilizar los comandos `\begin{figure}` y `\end{figure}` para indicarle que vamos a insertar una figura.

Después, se coloca la posición de la imagen con `\begin{figure}[H]` y colocando entre los corchetes la posición en la que queremos la imagen.

Seguidamente, se centra la imagen con el comando `\centering`. Posteriormente, vamos a indicar la proporción de la imagen respecto al texto y la ruta donde esta la imagen

`\includegraphics[proporcion de la imagen]{ruta de la imagen}`.

Y finalmente, cerramos la imagen con `\end{figure}`.

Por otro lado, podemos indicar las posiciones con los siguientes comandos:

- **h**: Le indicamos que ponga la imagen aquí.
- **t**: Le indicamos que lo ponga en la parte superior de la página.
- **b**: Le indicamos que lo ponga en la parte inferior de la página.
- **p**: Le indicamos que coloque los objetos flotantes en una página.
- **!**: Le indicamos que ignore las reglas internas de posicionamiento.
- **H**: Le indicamos que ponga la imagen aquí y para utilizarlo será necesario definir en el preámbulo `\usepackage{float}`.

## 2.4. Texto con color

Para que  $\text{\LaTeX}$  nos permita utilizar colores dentro de los documentos tendremos que escribir en el preámbulo los paquetes `\usepackage{color}` o `\usepackage{xcolor}`. Una vez los tenemos instalados, si queremos dar color al texto usaremos el comando `\color{}`, es muy importante poner entre corchetes la palabra a la que queremos darle el color, sino se coloreará todo el texto. Para ver cómo aplicar los colores vamos a poner el siguiente ejemplo:

Texto en color rojo.  
Palabra en color azul.

Listing 2.5: Colores 1

```
1 \usepackage{color}, importante definir previamente antes de
   utilizar el comando
2 \usepackage{xcolor}, podemos definir cualquiera de los dos en el
   preámbulo
3 \begin{center}
4 {\color{red}Texto en color rojo}.\
5 {\color{blue}Palabra} en color azul.
6 \end{center}
```

Como podemos observar, con estas herramientas podemos destacar texto con mucha facilidad, ya sea, cambiándole el color a la palabra, al texto o al fondo.

Por último, vamos a ver cómo definir colores. Para ello, primero debemos definir el paquete `\usepackage[usenames, dvipsnames]{color}` en el preámbulo y posteriormente, definiremos el color con el comando `\definecolor{nombrecolor}{RGB,rgb o cmyk}{numero RGB,rgb o cmyk}`.

Para personalizar nuestro color podemos recurrir a páginas web como <https://rgbcolorcode.com>. Ahora vamos a explicar cómo seleccionar el color:

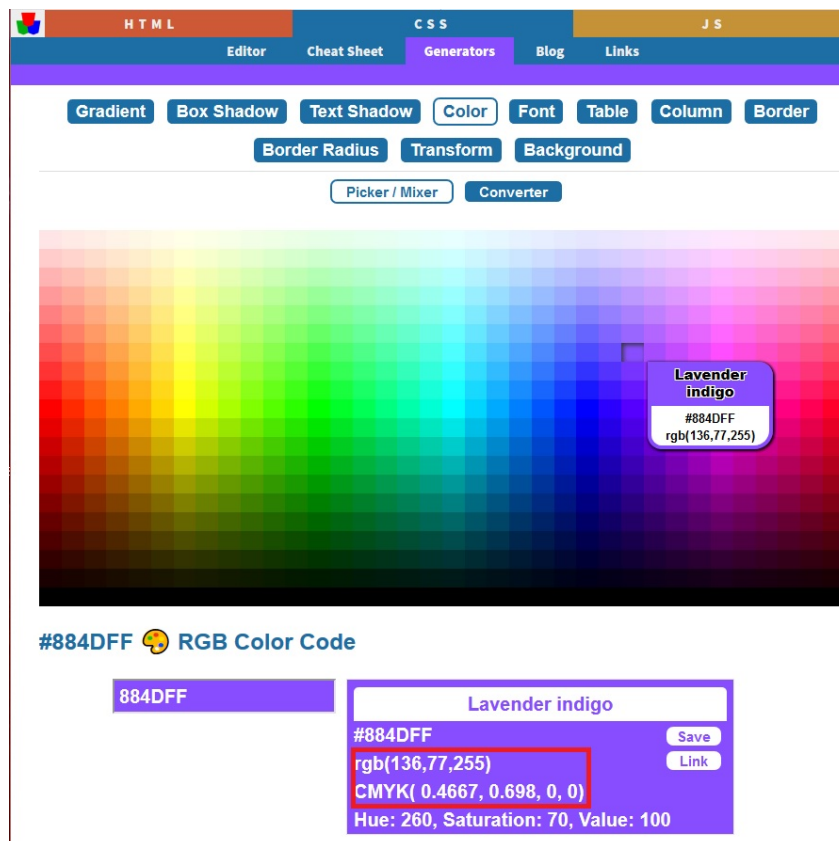


Figura 2.8: Página selección color

En la tabla de arriba, seleccionamos nuestro color, abajo lo podemos personalizar y en la tabla de al lado, tendremos los valores RGB y cmyk para poner en nuestro color. Ahora, con todo lo que hemos explicado, vamos a poner un ejemplo:

Texto con rojo personalizado.  
Texto con fondo azul.

Listing 2.6: Colores personalizados

```

1 \usepackage{color}, importante definir previamente antes de
  utilizar el comando
2 \usepackage{xcolor}, podemos definir cualquiera de los dos en el
  preambulo
3 \definecolor{miazul}{RGB}{0,255,255}
4 \definecolor{mirojo}{RGB}{255,42,0}
5 \begin{center}
6 \textcolor{mirojo}{Texto con rojo personalizado}.\
7 \colorbox{miazul}{Texto con fondo azul.}
8 \end{center}
    
```

**Importante!!**

Estos ejercicios no sirven si no te lo pasas bien. Escribe aquello que quieras, por muy raro que sea. Se aprende mejor cuando tienes humor.

**Ejercicio 1.** Modifica con geometría el texto para crear un A0 con un texto en medio.

**Ejercicio 2.** Crea un monstruo con geometría. Pon el texto en medio de un folio y cambia los márgenes superior, inferior y de los laterales para aplastar lo máximo el texto hacia el medio.

**Ejercicio 3.** GATOS, QUIERO VER GATOS EN EL DOCUMENTO, PONED FOTOS DE GATOS EN EL DOCUMENTO. (para ello usad los flotantes, la centralización y los captions)

**Ejercicio 4.** Escribe las ventajas de una nutrición basada en cocos en una tabla con el generador de tablas

**Ejercicio 5.** Vamos a escribir el enunciado de un ejercicio aleatorio y un frame para poner la solución.

**Ejercicio 6.** Realiza un frame, coloca dentro un gato, después coloca el gato también fuera.

**Ejercicio 7.** Coloca 2 gatitos en columna pero sin usar columnación.

**Ejercicio 8\*.** Esto es una tabla básica, intenta imitar la misma sin usar el generador:

Columna 1	Columna 2	Columna 3
Dato 1	Dato 2	Dato 3

# Capítulo 3

# Capítulo 3

## 3.1. Poner una página en landscape

Para poner una página en landscape primero, definiremos en el preámbulo el paquete `lscape`, y posteriormente, ya podremos utilizar el entorno.

Listing 3.1: Código landscape

```
1 \usepackage{lscape}
2
3 \begin{landscape}
4
5 \end{landscape}
```

## 3.2. Paquete multicol

Otra forma de poner columnas es mediante el paquete `multicol`. Una vez que lo nombramos en el preámbulo creamos el entorno `multicols`, para ello abrimos el entorno con `\begin{multicol}{número}` y cerramos el entorno con `\end{multicol}`. Este entorno tiene las siguientes características:

- Número de columnas.
- **Texto de título:** Se puede colocar entre corchetes.

Vamos a poner un ejemplo para ilustrar este apartado:

## Ejemplo multicol

David Pacios

14 de febrero de 2019

### 1. Ejemplo sección

Este texto se va a colocar en dos co- Se puede distribuir de muchas formas.  
lumnas.

Figura 3.1: Ejemplo de texto en multicol

Listing 3.2: Código paquete multicol

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage [ spanish ] { babel }
4 \usepackage { multicol }
5 \title { Ejemplo multicol }
6 \author { David Pacios }
7 \begin { document }
8 \maketitle
9 \section { Ejemplo sección }
10 \begin { multicol } { 2 }
11 \noindent
12 Este texto se va a colocar en dos columnas. \\
13 Se puede distribuir de muchas formas .
14 \end { multicol }
15 \end { document }
```

### 3.2.1. Separación de columnas

Además de dar formato a las columnas, podemos darle una cierta separación con el comando `\columnsep` dentro del comando `\setlength` para determinar la separación y entre corchetes, ponemos la separación que queremos.

Para tenerlo más claro ponemos un ejemplo:



## Ejemplo multicol

David Pacios

14 de febrero de 2019

### 1. Ejemplo sección

Este texto se va a colocar en  
dos columnas.

Se puede distribuir de muchas  
formas.

Figura 3.2: Ejemplo de texto en multicol con separación

Listing 3.3: Código paquete multicol con separación

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3
4 \usepackage [ spanish ] { babel }
5 \usepackage { multicol }
6 \setlength { \columnsep } { 3 cm }
7 \title { Ejemplo multicol }
8 \author { David Pacios }
9
10
11 \begin { document }
12 \maketitle
13 \section { Ejemplo sección }
14 \begin { multicol } { 2 }
15 \noindent
16 Este texto se va a colocar en dos columnas. \\
17 Se puede distribuir de muchas formas .
18 \end { multicol }
19 \end { document }
```

### 3.2.2. Columnas des-balanceadas

Cuando queremos quitar el formato estándar de distribución de texto utilizamos el `*`. Para tenerlo más claro vamos a poner un ejemplo:

## Ejemplo multicol

David Pacios

14 de febrero de 2019

### 1. Ejemplo sección

Este texto se va a colocar en dos columnas.

Se puede distribuir de muchas formas.

Podemos poner en texto distribuido de manera distinta en varias columnas. Aquí pone-

mos otra columna.

Y otra por aquí.

Y por otra.

Figura 3.3: Ejemplo de multicol sin formato estándar

Listing 3.4: Código paquete multicol sin formato estándar

```

1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3
4 \usepackage [ spanish ] { babel }
5 \usepackage { multicol }
6 \setlength { \columnsep } { 3 cm }
7 \title { Ejemplo multicol }
8 \author { David Pacios }
9 \begin { document }
10 \maketitle
11 \section { Ejemplo sección }
12
13 \begin { multicol* } { 2 }
14 \noindent
15 Este texto se va a colocar en dos columnas. \\
16 Se puede distribuir de muchas formas. \\
17 Podemos poner en texto distribuido de manera distinta en
    varias columnas.
18 \columnbreak
19 Aquí ponemos otra columna. \\
20 Y otra por aquí. \\
21 Y por otro.
22 \end { multicol* }
23 \end { document }

```

### 3.2.3. Delimitando con líneas verticales

Podemos colocar líneas verticales con colores para ello vamos a utilizar el comando `\def\columnseprulecolor{\color{color elegido}}`. Es importante definir el paquete `color` antes de añadirle el comando. Para ello, vamos a poner un ejemplo:

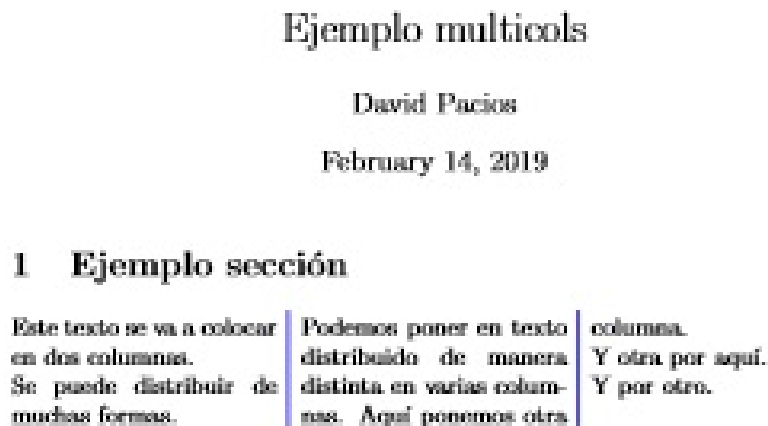


Figura 3.4: Ejemplo de multicol con líneas verticales azules

Listing 3.5: Código paquete multicol con líneas verticales azules

```

1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage [ english ] { babel }
4 \usepackage { multicol }
5 \usepackage { color }
6
7 \setlength { \columnseprule } { 1pt }
8 \def \columnseprulecolor { \color { blue } }
9
10 \title { Ejemplo multicol }
11 \author { David Pacios }
12
13 \begin { document }
14 \maketitle
15 \section { Ejemplo sección }
16 \begin { multicol } { 3 }
17 \noindent
18 Este texto se va a colocar en dos columnas. \\
19 Se puede distribuir de muchas formas. \\
20 Podemos poner en texto distribuido de manera distinta en
    varias columnas.

```

```

21 \columnbreak
22 Aquí ponemos otra columna.\\
23 Y otra por aquí.\\
24 Y por otro.
25 \end{multicols}
26 \end{document}

```

Por último, podemos definir el comando `\columnbreak`, que realiza una separación entre distintas columnas.

## 3.3. Matemáticas MMI

### 3.3.1. Modo matemático

Para entrar en el modo matemático es muy importante definir en el preámbulo el paquete `\usepackage{amsmath}`, ya que, sin este paquete no nos funcionará el modo matemático y nos dará error.

Una vez lo hayamos definido, nos permitirá utilizar todos los comandos para poder escribir nuestras fórmulas matemáticas. Seguidamente, vamos a listar todos los comandos que se pueden utilizar:

- `$`: Entrar y salir en el modo matemático en el modo texto. Es decir, que las fórmulas matemáticas están escritas dentro de un texto. Además de este comando, hay otro comando que se puede utilizar en el modo texto, que es el comando `\(\)`.
- `$$`: Entrar y salir del modo matemático resaltado, ya que, las funciones matemáticas están fuera del texto con un tamaño mayor. Además de este comando, hay otro comando que se puede utilizar para el modo de texto resaltado que es `\[ \]`.
- Otro entorno que se puede utilizar para escribir las ecuaciones que es el `equation`. Cuyos comandos son `\begin{equation}` y `\end{equation}`. De esta forma, también escribiremos nuestra fórmula de un modo resaltado, es decir, fuera del texto. Este modo nos permite enumerar nuestras ecuaciones, y en el caso de que no queramos siempre podemos añadirle el símbolo `*` al principio del comando.

Posteriormente, después de describir los comandos vamos a ver una serie de ejemplos para ver cómo se escriben las fórmulas:

La función de la recta pendiente es  $y = mx + b$  tangente a la recta.

Listing 3.6: Ejemplo de función matemática con el texto

```

1 La funcion de la recta pendiente es $y=mx+b$ tangente a la
  recta .

```

La función de la recta pendiente es  $y = mx + b$  tangente a la recta.

Listing 3.7: Ejemplo de función matemática con el texto resaltado

```
1 La funcion de la recta pendiente es \ (y=mx+b\ ) tangente a la
   recta .
```

La función de la recta pendiente es

$$y = mx + b$$

tangente a la recta.

Listing 3.8: Ejemplo de función matemática con el texto resaltado

```
1 La funcion de la recta pendiente es $$y=mx+b$$ tangente a la
   recta .
```

La función de la recta tangente es

$$y = mx + b$$

es tangente a la recta.

Listing 3.9: Ejemplo de función matemática con el texto resaltado

```
1 La funcion de la recta tangente es
2 \[
3 y=mx+b
4 \]
5 es tangente a la recta .
```

La función de la recta tangente es:

$$y = mx + b$$

es tangente a la recta.

Listing 3.10: Ejemplo de función matemática con el entorno de la ecuación sin numerar

```
1 La funcion de la recta tangente es:
2 \begin{equation*}
3 y=mx+b
4 \end{equation*}
5 es tangente a la recta .
```

La función de la recta tangente es

$$y = mx + b \tag{3.1}$$

es tangente a la recta.

Listing 3.11: Ejemplo de función matemática con el entorno de la ecuación numerado

```
1 La funcion de la recta tangente es
2 \begin{equation}
3 y=mx+b
4 \end{equation}
5 es tangente a la recta .
```

Como podemos ver, con este último entorno, se puede numerar la ecuación y si hacemos un índice de ecuaciones, nos aparecerá en este índice.

### 3.3.2. Tablas de símbolos básicos

Para poder usar los símbolos en  $\text{\LaTeX}$  es necesario escribir en el preámbulo el paquete `\usepackage{amssymb}`.

Una vez escrito, podemos escribir los siguientes símbolos:

$\alpha$ : <code>\alpha</code>	$\theta$ : <code>\theta</code>	$\circ$ : <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</code>	$\Sigma$ : <code>\Sigma</code>	$\Psi$ : <code>\Psi</code>
$\Gamma$ : <code>\Gamma</code>	$\Lambda$ : <code>\Lambda</code>	$\Upsilon$ : <code>\Upsilon</code>	$\Omega$ : <code>\Omega</code>
$\Delta$ : <code>\Delta</code>	$\Xi$ : <code>\Xi</code>	$\Phi$ : <code>\Phi</code>	
$\Theta$ : <code>\Theta</code>	$\Pi$ : <code>\Pi</code>		

Cuadro 3.1: Letras griegas

$F$ : <code>\digamma</code>	$\varkappa$ : <code>\varkappa</code>
-----------------------------	--------------------------------------

Cuadro 3.2: Letras griegas AMS

$\leftarrow$ : <code>\leftarrow</code>	$\longleftarrow$ : <code>\longleftarrow</code>	$\uparrow$ : <code>\uparrow</code>
$\Leftarrow$ : <code>\Leftarrow</code>	$\Longleftarrow$ : <code>\Longleftarrow</code>	$\Uparrow$ : <code>\Uparrow</code>
$\rightarrow$ : <code>\rightarrow</code>	$\longrightarrow$ : <code>\longrightarrow</code>	$\downarrow$ : <code>\downarrow</code>
$\Rightarrow$ : <code>\Rightarrow</code>	$\Longrightarrow$ : <code>\Longrightarrow</code>	$\Downarrow$ : <code>\Downarrow</code>
$\leftrightarrow$ : <code>\leftrightarrow</code>	$\longleftrightarrow$ : <code>\longleftrightarrow</code>	$\Updownarrow$ : <code>\Updownarrow</code>
$\Leftrightarrow$ : <code>\Leftrightarrow</code>	$\Longleftrightarrow$ : <code>\Longleftrightarrow</code>	$\Updownarrow$ : <code>\Updownarrow</code>
$\mapsto$ : <code>\mapsto</code>	$\longmapsto$ : <code>\longmapsto</code>	$\nearrow$ : <code>\nearrow</code>
$\hookrightarrow$ : <code>\hookrightarrow</code>	$\hookrightarrow$ : <code>\hookrightarrow</code>	$\searrow$ : <code>\searrow</code>
$\leftharpoonup$ : <code>\leftharpoonup</code>	$\rightharpoonup$ : <code>\rightharpoonup</code>	$\swarrow$ : <code>\swarrow</code>
$\leftharpoondown$ : <code>\leftharpoondown</code>	$\rightharpoondown$ : <code>\rightharpoondown</code>	$\nwarrow$ : <code>\nwarrow</code>
$\rightleftharpoons$ : <code>\rightleftharpoons</code>	$\leadsto$ : <code>\leadsto</code>	$\rightarrow$ : <code>\rightarrow</code>

Cuadro 3.3: Símbolos flechas

$\dashrightarrow$ : <code>\dashrightarrow</code>	$\dashleftarrow$ : <code>\dashleftarrow</code>
$\Lleftarrow$ : <code>\Lleftarrow</code>	$\Lrightarrow$ : <code>\Lrightarrow</code>
$\leftarrowtail$ : <code>\leftarrowtail</code>	$\looparrowleft$ : <code>\looparrowleft</code>
$\leftrightharpoons$ : <code>\leftrightharpoons</code>	$\curvearrowleft$ : <code>\curvearrowleft</code>
$\circlearrowleft$ : <code>\circlearrowleft</code>	$\Lsh$ : <code>\Lsh</code>
$\upuparrows$ : <code>\upuparrows</code>	$\upharpoonleft$ : <code>\upharpoonleft</code>
$\downharpoonleft$ : <code>\downharpoonleft</code>	$\multimap$ : <code>\multimap</code>
$\leftrightsquigarrow$ : <code>\leftrightsquigarrow</code>	$\rightrightarrows$ : <code>\rightrightarrows</code>
$\rightleftarrows$ : <code>\rightleftarrows</code>	$\twoheadrightarrow$ : <code>\twoheadrightarrow</code>
$\rightarrowtail$ : <code>\rightarrowtail</code>	$\looparrowright$ : <code>\looparrowright</code>
$\rightleftharpoons$ : <code>\rightleftharpoons</code>	$\curvearrowright$ : <code>\curvearrowright</code>
$\circlearrowright$ : <code>\circlearrowright</code>	$\Rsh$ : <code>\Rsh</code>
$\downdownarrows$ : <code>\downdownarrows</code>	$\upharpoonright$ : <code>\upharpoonright</code>
$\downharpoonright$ : <code>\downharpoonright</code>	$\rightsquigarrow$ : <code>\rightsquigarrow</code>

Cuadro 3.4: Flechas AMS

$\nleftarrow$ : <code>\nleftarrow</code>	$\nrightarrow$ : <code>\nrightarrow</code>	$\nLeftarrow$ : <code>\nLeftarrow</code>
$\nrightarrow$ : <code>\nrightarrow</code>	$\nleftrightarrow$ : <code>\nleftrightarrow</code>	$\nleftrightarrow$ : <code>\nleftrightarrow</code>

Cuadro 3.5: Flechas negación AMS

$\pm$ : <code>\pm</code>	$\cap$ : <code>\cap</code>	$\diamond$ : <code>\diamond</code>	$\oplus$ : <code>\oplus</code>
$\mp$ : <code>\mp</code>	$\cup$ : <code>\cup</code>	$\triangleup$ : <code>\bigtriangleup</code>	$\ominus$ : <code>\ominus</code>
$\times$ : <code>\times</code>	$\uplus$ : <code>\uplus</code>	$\triangledown$ : <code>\bigtriangledown</code>	$\otimes$ : <code>\otimes</code>
$\div$ : <code>\div</code>	$\sqcap$ : <code>\sqcap</code>	$\triangleleft$ : <code>\triangleleft</code>	$\oslash$ : <code>\oslash</code>
$\ast$ : <code>\ast</code>	$\sqcup$ : <code>\sqcup</code>	$\triangleright$ : <code>\triangleright</code>	$\odot$ : <code>\odot</code>
$\star$ : <code>\star</code>	$\vee$ : <code>\vee</code>	$\triangleleft$ : <code>\lhd</code>	$\bigcirc$ : <code>\bigcirc</code>
$\circ$ : <code>\circ</code>	$\wedge$ : <code>\wedge</code>	$\triangleright$ : <code>\rhd</code>	$\dagger$ : <code>\dagger</code>
$\bullet$ : <code>\bullet</code>	$\setminus$ : <code>\setminus</code>	$\triangleleft$ : <code>\unlhd</code>	$\ddagger$ : <code>\ddagger</code>
$\cdot$ : <code>\cdot</code>	$\wr$ : <code>\wr</code>	$\triangleright$ : <code>\unrhd</code>	$\amalg$ : <code>\amalg</code>

Cuadro 3.6: Operadores binarios

$\sum$ : <code>\sum</code>	$\bigcap$ : <code>\bigcap</code>	$\bigodot$ : <code>\bigodot</code>
$\prod$ : <code>\prod</code>	$\bigcup$ : <code>\bigcup</code>	$\bigotimes$ : <code>\bigotimes</code>
$\coprod$ : <code>\coprod</code>	$\bigsqcup$ : <code>\bigsqcup</code>	$\bigoplus$ : <code>\bigoplus</code>
$\int$ : <code>\int</code>	$\bigvee$ : <code>\bigvee</code>	$\biguplus$ : <code>\biguplus</code>
$\oint$ : <code>\oint</code>	$\bigwedge$ : <code>\bigwedge</code>	$\infty$ : <code>\infty</code>

Cuadro 3.7: Operadores de tamaño variable

$\leq$ : <code>\leq</code>	$\geq$ : <code>\geq</code>	$\equiv$ : <code>\equiv</code>	$\models$ : <code>\models</code>
$\prec$ : <code>\prec</code>	$\succ$ : <code>\succ</code>	$\sim$ : <code>\sim</code>	$\perp$ : <code>\perp</code>
$\preceq$ : <code>\preceq</code>	$\succeq$ : <code>\succeq</code>	$\simeq$ : <code>\simeq</code>	$\mid$ : <code>\mid</code>
$\ll$ : <code>\ll</code>	$\gg$ : <code>\gg</code>	$\asymp$ : <code>\asymp</code>	$\parallel$ : <code>\parallel</code>
$\subset$ : <code>\subset</code>	$\supset$ : <code>\supset</code>	$\approx$ : <code>\approx</code>	$\bowtie$ : <code>\bowtie</code>
$\subseteq$ : <code>\subseteq</code>	$\supseteq$ : <code>\supseteq</code>	$\cong$ : <code>\cong</code>	$\Join$ : <code>\Join</code>
$\sqsubset$ : <code>\sqsubset</code>	$\sqsupset$ : <code>\sqsupset</code>	$\neq$ : <code>\neq</code>	$\smile$ : <code>\smile</code>
$\sqsubseteq$ : <code>\sqsubseteq</code>	$\sqsupseteq$ : <code>\sqsupseteq</code>	$\doteq$ : <code>\doteq</code>	$\frown$ : <code>\frown</code>
$\in$ : <code>\in</code>	$\ni$ : <code>\ni</code>	$\propto$ : <code>\propto</code>	$==$ : <code>==</code>
$\vdash$ : <code>\vdash</code>	$\dashv$ : <code>\dashv</code>	$\langle \langle$ : <code>\langle \langle</code>	$\rangle \rangle$ : <code>\rangle \rangle</code>
$\ddot{\cdot}$ : <code>\ddot{\cdot}</code>	$\cdot$ : <code>\cdot</code>		

Cuadro 3.8: Operadores de relación

$\text{,}$	$\text{;}$	$\text{:}$ : <code>\colon</code>	$\text{.}$ : <code>\ldotp</code>	$\text{.}$ : <code>\cdotp</code>
------------	------------	----------------------------------	----------------------------------	----------------------------------

Cuadro 3.9: Signos de puntuación

$\sqrt{x}$ : <code>\sqrt{x}</code>	$\sqrt[y]{x}$ : <code>\sqrt[y]{x}</code>
------------------------------------	--

Cuadro 3.10: Otros símbolos

Hay más símbolos de los que se han explicado arriba, si queréis buscar más sólo tenéis que buscarlos en cualquier página web especializada en  $\text{\LaTeX}$  como  $\text{\Share\LaTeX}$  y  $\text{\Overleaf}$ .

### 3.3.3. Fórmulas simples

Ahora vamos a explicar cómo poner unas fórmulas simples, que son aquellas que tienen multiplicaciones y divisiones. Además de estas fórmulas simples, vamos a ver cómo elevar algún número y cómo poner los subíndices.

Primero, vamos a poner una fórmula simple, como por ejemplo:

$$3 \times 2 = 6$$

$$10 \div 2 = 5$$

Listing 3.12: Ejemplo del código de fórmulas simples

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[
3 3\times 2=6
4 \]
5 \[
6 10\div 2=5
7 \]
```



Con esta sencilla estructura, podemos crear unas fórmulas matemáticas que incluyan: sumas, restas, multiplicaciones y divisiones.

Seguidamente, vamos a explicar cómo se elevan los números y cómo se ponen los subíndices. Para elevar los números vamos a utilizar el comando `^{\}`, y entre los corchetes ponemos el número que queremos elevar. Para verlo más claro, vamos a poner varios ejemplos:

$$5^2 = 25$$

Listing 3.13: Ejemplo del código de elevación de números

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 5^2=25
4 \]
```

$$x^x y = x^{xy}$$

Listing 3.14: Ejemplo del código de elevación de números

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 x^{xy}=x^{\{xy\}}
4 \]
```

Como podemos ver en este segundo ejemplo, es muy importante poner entre corchetes los números que tienen que estar elevados, sino, si hay dos números que se quieren elevar, sólo se elevará el primero y no el segundo.

Y por último, vamos a ver cómo poner los subíndices, para realizarlo, vamos a utilizar el comando `_{\}`, y colocando entre los corchetes el número que queremos colocar el subíndice. Para verlo en acción vamos a poner varios ejemplos:

$$x_1 + x_2 = x_3$$

Listing 3.15: Ejemplo del código de subíndice de números

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 x_1+x_2=x_3
4 \]
```

$$y_{xyz} + x_{xyz} + z_{xyz} = 2xy_zx_yz$$

Listing 3.16: Ejemplo del código de subíndice de números

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[
3 y_{xyz}+x_{xyz}+z_{xyz}=2xyz_{xyz}
4 \]
```

Como podemos ver, al igual que al hacer las elevaciones, si hay más de un número que queremos llevar al subíndice los tendremos que colocar entre los corchetes, sino nos pasará como en el ejemplo, que no todos los números se coloquen en el subíndice.

### 3.3.4. Estructuras matemáticas: límites, sumatorios

En este capítulo vamos a explicar dos tipos de estructuras matemáticas: los límites y los sumatorios.

Primero vamos a empezar con los límites cuyo comando es `\lim_{x\to}`. En el subíndice vamos a colocar hacia donde tiene el límite y arriba colocaremos nuestro límite. Y su comportamiento es diferente dependiendo del modo matemático donde estemos. Para visualizarlo, vamos a poner diferentes ejemplos:

$$\lim_{x \rightarrow 0} \frac{1}{x^2} = \infty$$

Listing 3.17: Ejemplo del código de límites

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[
3 \lim_{x\to 0}\dfrac{1}{x^2}=\infty
4 \]
```

$$\lim_{x \rightarrow 0} \frac{1}{x^2} = \infty$$

Ejemplo del código de límites

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \begin{center}
3 $\lim_{x\to 0}\dfrac{1}{x^2}=\infty$
4 \end{center}
```

$$\lim_{x \rightarrow 0} \frac{1}{x^2} = \infty$$

Listing 3.18: Ejemplo del código de límites

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \begin{equation*}
3 \lim_{x\to 0}\dfrac{1}{x^2}=\infty
4 \end{equation*}

```

Como podemos ver, si estamos en distintos modos matemáticos, los límites no se van a comportar de la misma manera, ya que, cambia la posición del subíndice y el ancho del límite. Además, podemos ver que siguen una estructura muy sencilla que es muy fácil de seguir.

Seguidamente, vamos a explicar cómo se realizan los sumatorios. Para realizarlo, utilizamos el comando `\sum_{n=1}^{\infty}`, y su tamaño depende del modo matemático que estemos utilizando. Para tenerlo más claro, vamos a poner varios ejemplos con todos los modos matemáticos para ver cómo se comporta:

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

Listing 3.19: Ejemplo del código de sumatorio

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \begin{center}
3 $\sum_{n=1}^{\infty} 2^{-n}=1$
4 \end{center}

```

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

Listing 3.20: Ejemplo del código de sumatorio

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[
3 \sum_{n=1}^{\infty} 2^{-n}=1
4 \]

```

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

Listing 3.21: Ejemplo del código de sumatorio

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \begin{equation*}
3 \sum_{n=1}^{\infty} 2^{-n}=1
4 \end{equation*}

```

Como podemos observar, el sumatorio no se comporta igual en todos los modos matemáticos. Por ejemplo, si queremos cambiar la posición de los subíndices o los superíndices en el modo texto, vamos a utilizar el comando `\limits` a continuación de la operación. Vamos a aplicar este comando al ejemplo anterior:

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

Listing 3.22: Ejemplo del código de sumatorio con los subíndices cambiados

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \begin{center}
3 $\sum\limits_{n=1}^{\infty} 2^{-n}=1$
4 \end{center}

```

Por el contrario, si queremos que los subíndices se hagan a un lado en el modo texto resaltado vamos a utilizar el comando `\nolimits`. Vamos a aplicar este comando nuevo al ejemplo:

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

Listing 3.23: Ejemplo del código de sumatorio con los subíndices a un lado

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 \sum\nolimits_{n=1}^{\infty} 2^{-n}=1
4 \]

```

Como podemos ver, los sumatorios no se comportan igual en todos los modos matemáticos, esto se puede aplicar igual a los operadores de tamaño variable y a funciones como son los límites.

Los operadores de tamaño variable se pueden encontrar en el capítulo de símbolos matemáticos.

### 3.3.5. Estructuras matemáticas: fracciones y raíces

En este capítulo vamos a explicar las fracciones y las raíces, tienen estructuras sencillas, pero si queremos añadirle algún elemento tipo paréntesis su estructura cambia y además, también cambian según el modo matemático en el que estemos.

Primero vamos a explicar las fracciones, las cuales podemos expresar por dos comandos, el comando `\frac{numerador}{denominador}` y `\dfrac{numerador}{denominador}`. La diferencia que hay entre estos dos comandos es que en el comando `\frac{numerador}{denominador}`, la fracción va estar expresada de una manera estándar y el comando `\dfrac{numerador}{denominador}` expresa la fracción en el modo `displaystyle`, este modo disminuye un poco el tamaño de la fracción respecto al anterior. Para tener más clara esta diferencia vamos a verlo con varios ejemplos:

$$\frac{1}{2}x = \frac{1}{2}x$$

Listing 3.24: Ejemplo del código de los dos comandos de fracciones

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 \dfrac{1}{2}x=\frac{1}{2}x
4 \]
```

Además de lo anterior, la fracción cambia de tamaño según el modo matemático en el que estemos, ya que, no se va a representar en el modo texto que en el resaltado. Para verlo de una manera más clara vamos a poner un ejemplo que contenga una fracción en estos dos modos:

La fracción  $\frac{1}{2}x$  es igual a:

$$\frac{1}{2}x$$

Y esta última es de una envergadura similar a:

$$\frac{1}{2}x$$

Listing 3.25: Ejemplo del código de la fracción en los distintos entornos matemáticos

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 La fraccion $\frac{1}{2}x$ es igual a:
3 \[
4 \frac{1}{2}x
5 \]
6 Y esta ultima es de un envergadura similar a:
7 \begin{equation*}
8 \frac{1}{2}x
9 \end{equation*}
```

Como podemos observar, el tamaño de la fracción no es el mismo en los distintos modos. Este cambio de tamaño en los distintos modos es más fácil de ver con el comando `\frac{numerador}{denominador}` que con el comando `\dfrac{numerador}{denominador}`. Además de tener estos problemas de tamaño según el modo en el que estemos, también nos encontramos con otro problema, el tamaño de los paréntesis o corchetes no se corresponde con nuestra fracción. Para verlo vamos a poner un ejemplo:

$$\left(\frac{1}{2}x\right)^2 = \frac{1}{4}x^2$$

Listing 3.26: Ejemplo del código del tamaño desigual de los delimitadores

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 \left(\dfrac{1}{2}x\right)^2=\dfrac{1}{4}x^2
4 \]
```

Cuando nos encontremos con este problema, tenemos dos opciones:

- Escribir al inicio del delimitador el comando `\left` y al final del delimitador el comando `\right` para que se ajuste automáticamente al tamaño de la fracción.
- Por otro lado, podemos poner el comando `\Big` para hacer más grande el delimitador.

Para ver como actúan los dos comandos vamos a corregir el ejemplo anterior:

$$\left(\frac{1}{2}x\right)^2 = \frac{1}{4}x^2$$

Listing 3.27: Ejemplo del código corrigiendo el tamaño de los delimitadores con `left` y `right`

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 \left(\dfrac{1}{2}x\right)^2=\dfrac{1}{4}x^2
4 \]
```

$$\left(\frac{1}{2}x\right)^2 = \frac{1}{4}x^2$$

Listing 3.28: Ejemplo del código corrigiendo el tamaño de los delimitadores con `Big`

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 \Big(\dfrac{1}{2}x\Big)^2=\dfrac{1}{4}x^2
4 \]
```

Lo que podemos observar de los ejemplos anteriores es que el comando `\Big` no se ajusta igual que los comandos `\left` y `\right` a la fracción, ya que con el comando no es capaz de ajustar su tamaño al de la fracción. El usuario tendrá que decir que tipo de comando `\Big` quiere ajustar a su fracción.

Por otro lado, tenemos las raíces que pueden ser expresadas por los comandos `\sqrt{}` o `\sqrt[ ]{}`. Depende de qué tipo de raíz estemos haciendo elegir un comando u otro, ya que, si vamos a hacer una raíz cuadrada elegiremos el comando `\sqrt{}` o si vamos a hacer una raíz cúbica o cuarta vamos a escoger el comando `\sqrt[ ]{}`. Para ver más clara la diferencia entre estos dos comandos vamos a realizar dos ejemplos:

$$\sqrt{2} \text{ es la raíz cuadrada de 2.}$$

$$\sqrt[4]{2} \text{ es la raíz cuarta de 2.}$$

Listing 3.29: Ejemplo del código de la diferencia entre los comandos de las raíces

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
```

```

2 \begin{center}
3 $\sqrt{2}$ es la raiz cuadrada de 2.\\
4 $\sqrt[4]{2}$ es la raiz cuarta de 2.
5 \end{center}

```

Como podemos ver, una vez sabemos el código de las fracciones y las raíces podemos expresarla de una manera muy sencilla.

### 3.3.6. Estructuras matemáticas: integrales

En este capítulo vamos a explicar cómo realizar todo tipo de integrales, desde las simples pasando por las integrales en dos puntos y terminando con las dobles.

Lo primero que tenemos que tener en cuenta es el comando de la integral que es `\int`, ya sólo con esto seremos capaces de realizar una integral inmediata como la del siguiente ejemplo:

$$\int x^6 dx = \frac{x^7}{7} + C$$

Listing 3.30: Ejemplo del código de una integral inmediata

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[
3 \int x^6 dx=\dfrac{x^7}{7}+C
4 \]

```

Una vez tengamos claro como hacer una integral inmediata, podemos pasar a realizar una integral en dos puntos. Para ello vamos a utilizar el siguiente comando `\int_{P1}^{P2}` para delimitarla entre dos puntos. Para tenerlo más claro vamos a poner un ejemplo:

$$\int_0^2 (2x + 2 - x^2 - 2) dx = \left[ x^2 - \frac{x^3}{3} \right]_0^2 = 4 - \frac{8}{3} = \frac{4}{3}$$

Listing 3.31: Ejemplo del código de una integral en dos puntos

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[
3 \int_{0}^{2} (2x+2-x^2-2) dx=\left[ x^2-\dfrac{x^3}{3} \right]_{0}
   ^{2}=4-\dfrac{8}{3}=\dfrac{4}{3}
4 \]

```

Seguidamente, si queremos realizar integrales dobles y triples utilizaremos comandos como `\iint` y `\iiint`. Para ver cómo serían estas integrales vamos a poner dos ejemplos:

$$\iint f(x) = \iiint g(x)$$

Listing 3.32: Ejemplo del código de las integrales dobles y triples

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 \iint f(x)=\iiint g(x)
4 \]

```

Además, si queremos juntar una integral junto con algún elemento más como una fracción, no quedará de todo bien. Como lo podemos ver en el siguiente ejemplo:

$$x = \frac{\int_x^y f(x)dx}{g(x)}$$

Listing 3.33: Ejemplo del código de la integral junto con una fracción

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 x=\dfrac{\int_{x}^{y} f(x) dx}{g(x)}
4 \]

```

Si queremos solucionar este problema, utilizaremos el comando `\displaystyle` para que la integral esté bien dispuesta en la fracción. Con esto, vamos a solucionar este ejemplo:

$$x = \frac{\int_x^y f(x)dx}{g(x)}$$

Listing 3.34: Ejemplo del código de la integral junto con una fracción con la integral bien dispuesta

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 x=\dfrac{\displaystyle \int_{x}^{y} f(x) dx}{g(x)}
4 \]

```

Y por último, si queremos realizar integrales cerradas simples. Y para representarlas utilizaremos el comando `\oint`. Para verlo más claro vamos a hacer un ejemplo:

$$\oint_x = \int x$$

Listing 3.35: Ejemplo del código de la integral cerrada simple

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 \oint_x=\int x
4 \]

```



### 3.3.7. Estructuras matemáticas: matrices, determinantes

Para la realización de matrices y determinantes de matrices vamos a utilizar los comandos `\begin{array}{c}` y `\end{array}`, y entre estos comandos se encontrarán los delimitadores. Para ver cómo se forma una matriz vamos a realizar un ejemplo:

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

Listing 3.36: Ejemplo del código de una matriz

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[A=\left(\begin{array}{cc}
3 1 & 2 \\
4 2 & 1
5 \end{array}\right)
6 \]
```

Las `{ccc}` representan las columnas que tiene la matriz, si tiene por ejemplo, dos columnas se representará como `{cc}`. Por tanto, las `c` representan el número de columnas.

Otra forma de realizar matrices es mediante el comando `\bmatrix`. Este comando nos permite poner matrices entre corchetes. De esta manera tenemos el siguiente ejemplo:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Listing 3.37: Ejemplo del código de matriz entre corchetes

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[I=\begin{bmatrix}
3 1 & 0 & 0 \\
4 0 & 1 & 0 \\
5 0 & 0 & 1
6 \end{bmatrix}
7 \]
```

Por otro lado, tenemos el comando `\pmatrix`, que nos permite poner a la matriz entre paréntesis. Para ver cómo se usa vamos a realizar un ejemplo:

$$M = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 6 \end{pmatrix}$$

Listing 3.38: Ejemplo del código de matriz entre paréntesis

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
```

```

2 \[M=\begin{pmatrix}
3 1 & 2 & 3 & 4 & 5 \\
4 6 & 7 & 8 & 9 & 6 \\
5 \end{pmatrix}
6 \]

```

Dentro del entorno de la matriz encontramos dos elementos a destacar:

- `&`: Para separar las columnas de la matriz.
- `\\`: Para separar filas.

Además de poder realizar matrices, podemos realizar operaciones con matrices como multiplicaciones, divisiones... Además, podemos combinar distintos entornos de matrices como podemos ver en el siguiente ejemplo:

$$A \cdot B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

Listing 3.39: Ejemplo de código de operaciones entre matrices

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[A \cdot B = \begin{pmatrix}
3 1 & 0 \\
4 0 & 1 \\
5 \end{pmatrix} \cdot \left( \begin{array}{cc}
6 2 & 0 \\
7 0 & 2 \\
8 \end{array} \right) = \begin{pmatrix}
9 2 & 0 \\
10 0 & 2 \\
11 \end{pmatrix}
12 \]

```

Por otro lado, para representar los determinantes de estas matrices cambiaremos los delimitadores a `\left|` y `\right|` para cambiar el paréntesis a este delimitador. Para ver el cambio de la matriz al determinante vamos a poner un ejemplo:

$$|A| = \begin{vmatrix} 1 & 2 \\ 2 & 1 \end{vmatrix}$$

Listing 3.40: Ejemplo del código de un determinante

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[ |A|=\left| \begin{array}{cc}
3   1 & 2 \\
4   2 & 1 \\
5 \end{array} \right|
6 \]
```

### 3.3.8. Estructuras matemáticas: textificación

En este capítulo vamos a ver cómo poner texto en nuestras ecuaciones y otros tipos de construcciones que podemos realizar en el modo matemático.

Primero, para colocar texto en nuestro modo matemático usaremos el comando `\text{}`, si no ponemos este comando, el texto saldrá mal. Para verlo más claro vamos a poner un ejemplo:

$$y = mx + b \quad \text{es la ecuación de la recta}$$

Listing 3.41: Ejemplo del código de un texto dentro del modo matemático

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 y=mx+b\quad\text{es la ecuacion de la recta}
4 \]
```

Como podemos ver en el ejemplo, con el comando nuestro texto se coloca perfectamente y con su espaciado, como si estuviera fuera del modo matemático.

Por otro lado, vamos a ver ciertas construcciones que nos pueden ser útiles para el modo matemático. Entre estas construcciones destacan el `\overbrace{Texto o función}` que pone un brazo por encima del texto y `\underbrace{Texto o función}` colocará un brazo por debajo del texto o de la función. Además, si queremos poner algo por debajo del brazo sólo tendremos que añadirle el comando `_` o si queremos ponerle algo por arriba del brazo superior tendremos que añadirle el comando `^`. Para verlo más claro vamos a poner varios ejemplos:

$$\overbrace{x^2 + x}^{x(x+1)} = \underbrace{x^2 + x}_{x(x+1)}$$

Listing 3.42: Ejemplo del código de brazo superior y brazo inferior

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
  formula
2 \[
3 \overbrace{x^2+x}^{x(x+1)}=\underbrace{x^2+x}_{x(x+1)}
4 \]
```

Como podemos ver, con estos brazos podemos añadir lo que queramos abajo o arriba de la función o el texto. Además, podemos combinar estos brazos con textos. Vamos a realizar un ejemplo con esta combinación:

$$\overbrace{x^2+x}^{x(x+1)} = \underbrace{x^2+x}_{\text{Lo mismo que en el anterior}}$$

Listing 3.43: Ejemplo del código de brazo superior y brazo inferior con texto

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[
3 \overbrace{x^2+x}^{x(x+1)}=\underbrace{x^2+x}_{\text{Lo mismo
   que en el anterior}}
4 \]
```

Estas expresiones se pueden concatenar de la siguiente manera: `\underbrace{Texto}_\underbrace{Texto}`. Para tenerlo más claro vamos a poner un ejemplo en el que se concatenen varias de estas expresiones:

$$\overbrace{x^2+x}^{\underbrace{x(x+1)}_1}$$

$$\overbrace{\overbrace{\text{Texto cualquiera}}}$$

$$\underbrace{\underbrace{\text{Texto cualquiera}}}$$

Listing 3.44: Ejemplo del código de brazo superior y brazo inferior con texto y concatenados

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[
3 \underbrace{x^2+x}_{\underbrace{x(x+1)}_1}
4 \]
5 \[
6 \overbrace{\text{Texto cualquiera}}^{\overbrace{\text{Texto
   cualquiera}}}
7 \]
```

Otra construcción que podemos ver es el vector que utilizará el comando `\Vec{}`, y entre los corchetes pondremos nuestro vector. Para verlo mejor vamos a poner un ejemplo:

$$\vec{x}_1 = \vec{x}_2 - \vec{x}_3$$

### 3.3.9. Estructuras matemáticas: Sistemas de ecuaciones

Hay varias formas de representar los sistemas de ecuaciones, una es mediante el entorno `array` y otra es mediante el entorno `eqnarray` donde las ecuaciones se nos representan respecto al `=`.

Primero vamos a ver cómo se representan las ecuaciones mediante el entorno `array`, y para ello, vamos a utilizar los comandos `\begin{array}{ll}` donde las `ll` representan el número de ecuaciones y terminando con el comando `\end{array}`. Separaremos las ecuaciones mediante el comando `\\`. Además, interpondremos una llave mediante el comando `\left\{` si la llave está a la izquierda o mediante el comando `\right\}` si está a la derecha, en el caso que esté la llave a uno de los lados, en el otro lado se usará el comando `\left.` o `\right..` Para tenerlo más claro, vamos a poner un ejemplo, uno con la llave a la izquierda y otra con la llave a la derecha:

$$\left\{ \begin{array}{l} x + y + 2z = 45 \\ 2x + 4y + 18z = 145 \\ 47x + 23y + 89z = 1089 \end{array} \right.$$

Listing 3.45: Ejemplo del código de sistemas de ecuaciones con llave a la izquierda

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \left\{\begin{array}{l}
3 x+y+2z=45\\
4 2x+4y+18z=145\\
5 47x+23y+89z=1089
6 \end{array}\right.
7 \}

```

$$\left. \begin{array}{l} 10x + 20y + 80z = 2000 \\ 2x + 8y + 50z = 45 \\ 11x + 44y + 63z = 78 \end{array} \right\}$$

Listing 3.46: Ejemplo del código de sistemas de ecuaciones con llave a la derecha

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \left.\begin{array}{l}
3 10x+20y+80z=2000\\
4 2x+8y+50z=45\\
5 11x+44y+63z=78
6 \end{array}\right\}
7 \}

```

Además de poder darle estos formatos, también podemos realizar el sistema de ecuaciones sin llaves. Para ello, utilizaremos los comandos `\left.` y `\right..` Y vamos a demostrarlo

con un ejemplo:

$$\begin{aligned}x + y + z &= 10 \\2x + 45y + 14z &= 89 \\588x + 789y + 123z &= 1023 \\89x + 74y + 45z &= 630\end{aligned}$$

Listing 3.47: Ejemplo del código de sistemas de ecuaciones sin llave

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \[ \left. \begin{array}{l}
3 x+y+z=10 \\
4 2x+45y+14z=89 \\
5 588x+789y+123z=1023 \\
6 89x+74y+45z=630
7 \end{array} \right. \right.
8 \]
```

Por otro lado, tenemos el entorno `eqnarray` donde se nos colocará el sistema de ecuaciones con un `=` en medio y con la misma separación en todas las operaciones. Aquí no nos hace falta indicar el número de ecuaciones que vamos a poner, eso sí, para separar las ecuaciones vamos a utilizar el comando `\\` y para colocar las ecuaciones utilizaremos `&=&` para separar las ecuaciones. Para ilustrar todo lo anterior vamos a poner un ejemplo:

$$y = x \tag{3.2}$$

$$x^2 + 2y = 3x^2 + 2y \tag{3.3}$$

$$(x + y)(2x - y) = 3x - y \tag{3.4}$$

Listing 3.48: Ejemplo del código del entorno `eqnarray`

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \begin{eqnarray}
3 y&=&x \\
4 x^2+2y&=&3x^2+2y \\
5 (x+y)(2x-y)&=&3x-y
6 \end{eqnarray}
```

Y por último, a este entorno `eqnarray` se le puede añadir cajas sin marco como `\mbox{}` para colocar un texto en nuestro sistema de ecuaciones. Para verlo lo podemos colocar en el siguiente ejemplo:

$$y = x \tag{3.5}$$

$$x^2 + 2y = 3x^2 + 2y \text{(Ecuación a destacar)} \tag{3.6}$$

$$(x + y)(2x - y) = 3x - y \tag{3.7}$$

Listing 3.49: Ejemplo del código del entorno eqnarray con caja mbox

```

1 \usepackage{amsmath}, definir antes de utilizar cualquier
   formula
2 \begin{eqnarray}
3 y&=&x \\
4 x^2+2y&=&3x^2+2y \mbox{( Ecuacion a destacar )} \\
5 (x+y)(2x-y)&=&3x-y
6 \end{eqnarray}

```

### 3.4. Licenciar documentos

Para licenciar documentos vamos a utilizar dos paquetes, el `verbatim` para indicar la licencia al final y el `doclicense` para ajustar la licencia.

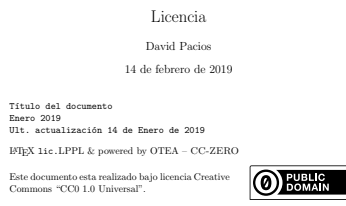


Figura 3.5: Documento con licencia

Listing 3.50: Código licencia

```

1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage [ spanish ] { babel }
4 \usepackage { verbatim }
5
6 \usepackage [

```

```
7     type={CC},
8     modifier={zero},
9     version={1.0},
10  ]{doclicense}
11
12  \title{Licencia}
13  \author{David Pacios}
14
15
16  \begin{document}
17
18  \maketitle
19
20  \begin{verbatim}
21  Título del documento
22  Enero 2019
23  Ult. actualización 14 de Enero de 2019
24  \end{verbatim}
25  \LaTeX{} {\verb lic. LPPL} \& powered by OTEA -- \textsc{CC-
26  ZERO}
27  \doclicenseThis
28  \end{document}
```

Con el comando `modifier` podremos modificar el tipo de licencia y con el comando `version` podremos modificar la versión. Es muy importante definir el paquete `verbatim` antes de compilar la licencia.

### 3.5. Paquete fancyhdr

Con este paquete podemos darle estilo a la página, podemos cambiar los encabezados y los pies de página.

Primero definiremos en el preámbulo el paquete `fancyhdr`, y seguidamente, en el mismo preámbulo si queremos poner el encabezado definiremos el comando `\fancyhead[]{}` y por otro lado, definiremos el comando `\fancyfoot[]{}`, y entre corchetes, pondremos lo siguiente:

- **E**: Página par.
- **O**: Página impar.
- **L**: Campo izquierdo.
- **R**: Campo derecho.
- **H**: Encabezado.
- **F**: Pie de página.



Y finalmente, si queremos cambiar o quitar la línea decorativa utilizaremos el comando `\renewcommand{\headrulewidth}{0pt}`.



The screenshot shows a LaTeX document with a horizontal line at the top. Below the line, the left side contains the text "Encabezados y pies de página" and the right side contains "Overleaf". Below this, there is a section header "1. Sección" followed by the text "Texto capítulo".

Página 1

Figura 3.6: Encabezado y pie de página

Listing 3.51: Código encabezado y pie de página

```
1 \documentclass { article }
2 \usepackage [ spanish ] { babel }
3 \usepackage [ utf8 ] { inputenc }
4 \usepackage { fancyhdr }
5 \pagestyle { fancy }
6 \fancyhf {}
7 \rhead { Overleaf }
8 \lhead { Encabezados y pies de página }
9 \rfoot { Página \thepage }
```

```

10
11 \begin{document}
12
13 \section{Sección}
14 \noindent
15 Texto capítulo
16
17 \end{document}

```

### 3.6. Paquete mdframed

El paquete `mdframed`, que es un paquete muy sencillo para encuadrar el texto y nos permite manejar encuadres de grandes dimensiones. Primero, definimos en el preámbulo el paquete `\usepackage{mdframed}`. Después, utilizamos los comandos `\begin{mdframed}` y `\end{mdframed}`. Ahora vamos a aplicar lo que hemos aprendido al siguiente ejemplo:

Texto con encuadre `mdframed` para tener un encuadre sencillo.

Listing 3.52: Caja mdframed

```

1 \usepackage{mdframed}, importante definir previamente antes de
  utilizar el comando
2 \begin{mdframed}
3 Texto con encuadre {\color{blue}mdframed} para tener un
  encuadre sencillo.
4 \end{mdframed}

```

Y por último, vamos a ver cómo personalizar la caja `mdframed`, que es la más sencilla de personalizar respecto a los anteriores. Para ello, sólo tenemos que poner en nuestro comando `\begin{mdframed}[backgroundcolor=yellow]`. Con sólo esto definido, podemos agregarle un color a nuestra caja. Para verlo vamos a poner un ejemplo:

Mdframed con fondo amarillo.

Listing 3.53: Tabla coloreada mdframed

```

1 \usepackage{mdframed}, definir antes de la caja
2 \begin{mdframed}[backgroundcolor=yellow]
3 Mdframed con fondo amarillo.
4 \end{mdframed}

```

**Importante!!**

Estos ejercicios no sirven si no te lo pasas bien. Escribe aquello que quieras, por muy raro que sea. Se aprende mejor cuando tienes humor.

**Ejercicio 1.** Escribe una ecuación en modo matemático y modo matemático extendido.

**Ejercicio 2.** Ahora que sabes usar mdframes, colores, y matemáticas de forma correcta. Inventa un enunciado gracioso como si fuera un ejercicio y haz la resolución con todos esos elementos anteriores.

**Ejercicio 3.** Realiza una fracción y una fracción con estilo.

**Ejercicio 4.** Licencia este documento como tú quieras.

**Ejercicio 5.** Realiza una cabecera bonita para tu documento, mezclado con una geometría de 0.7 tiene que quedar super profesional!



# Capítulo 4

# Capítulo 4

## 4.1. Símbolos en los itemizados

La personalización del entorno `itemize`, para ello tendremos que utilizar el comando `\renewcommand{\labelitem<n>}{<symbol>}`, en el primer corchete indicaremos el número de itemizado que va a utilizar el símbolo y en el segundo, el símbolo que vamos a utilizar.

### Itemizado personalizado

David Pacios

February 2019

- ◊ Un diamante.
- ◊ Dos diamantes.
- ◊ Tres diamantes.
- × Uno.
- × Dos.
- × Tres.
  - Uno.
  - Dos.

Figura 4.1: Itemizados personalizados

Listing 4.1: Código personalización itemize

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3
4 \title { Itemizado personalizado }
5 \author { David Pacios }
6 \date { February 2019 }
7
8 \begin { document }
9 \maketitle
10 \renewcommand { \labelitemi } { $ \diamond $ }
11 \begin { itemize }
12     \item Un diamante .
13     \item Dos diamantes .
14     \item Tres diamantes .
15 \end { itemize }
16 \renewcommand { \labelitemi } { $ \times $ }
17 \renewcommand { \labelitemii } { $ \bullet $ }
18 \renewcommand { \labelitemiii } { $ \checkmark $ }
19 \begin { itemize }
20     \item Uno .
21     \item Dos .
22     \item Tres .
23 \begin { itemize }
24     \item Uno .
25     \item Dos .
26 \end { itemize }
27 \end { itemize }
```

Como podemos ver, en la parte del código `\labelitemi` se escribirá en números romanos el número de itemizado que queramos modificar.

Y por otro lado, podemos usar el paquete `\usepackage{pifont}` para poder utilizar en el comando `\renewcommand{\labelitemi}{\ding{}}`, y en ese `ding` colocaremos los siguientes símbolos de la siguiente tabla:

	0	1	2	3	4	5	6	7
0								
8								
16								
24								
32		✂	✂	✂	✂	✂	✂	✂
40	✂	✂	✂	✂	✂	✂	✂	✂
48	✂	✂	✂	✂	✂	✂	✂	✂
56	✂	✂	✂	✂	✂	✂	✂	✂
64	✂	✂	✂	✂	✂	✂	✂	✂
72	✂	✂	✂	✂	✂	✂	✂	✂
80	✂	✂	✂	✂	✂	✂	✂	✂
88	✂	✂	✂	✂	✂	✂	✂	✂
96	✂	✂	✂	✂	✂	✂	✂	✂
104	✂	✂	✂	✂	●	○	■	□
112	□	□	□	▲	▼	◆	◇	◐
120				‘	’	“	”	
128								
136								
144								
152								
160		♣	♣	♣	♣	♣	♣	♣
168	♣	♣	♣	♣	①	②	③	④
176	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫
184	⑬	⑭	⑮	⑯	⑰	⑱	⑲	⑳
192	①	②	③	④	⑤	⑥	⑦	⑧
200	⑨	⑩	①	②	③	④	⑤	⑥
208	⑦	⑧	⑨	⑩	→	→	↔	↕
216	↖	→	↗	→	→	→	→	→
224	→	→	→	→	→	→	→	→
232	→	→	→	→	→	→	→	→
240	→	→	→	→	→	→	→	→
248	→	→	→	→	→	→	→	→

Figura 4.2: Tabla para utilizar en el ding

Esta tabla ha sido obtenida de la web <http://willbenton.com/wb-images/pifont.pdf> donde podremos encontrar todos los símbolos y qué número poner. Para tenerlo más claro vamos a poner un ejemplo:

Itemizado personalizado

David Pacios  
February 2019

☛ Un símbolo.

☛ Dos símbolos.

☛ Uno.

☛ Dos.

Figura 4.3: Ejemplo de itemizado con ding

Listing 4.2: Código paquete pifont

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage { pifont }
4 \title { Itemizado personalizado }
5 \author { David Pacios }
6 \date { February 2019 }
7
8 \begin { document }
9
10 \maketitle
11
12 \renewcommand { \labelitemi } { \ding { 43 } }
13 \renewcommand { \labelitemii } { \ding { 42 } }
14 \begin { itemize }
15     \item Un simbolo .
16     \item Dos simbolos .
17 \begin { itemize }
18     \item Uno .
19     \item Dos .
20 \end { itemize }
21 \end { itemize }
22 \end { document }
```

## 4.2. Letras en los enumerados

Por otro lado, tenemos el entorno `enumerate`, en el cual podemos poner letras sólo tenemos que utilizar al igual que en el comando anterior, el comando `\renewcommand{\theenumii}{\Roman{enumi}}` para seleccionar que lo haga en números romanos. Este segundo parámetro se puede modificar de la siguiente forma:

- **roman**: Números romanos en minúsculas.
- **Alph**: Alfabético en mayúsculas.
- **alph**: Alfabético en minúsculas.

Figura 4.4: Letra en un enumerado

Listing 4.3: Código de enumerado con letra

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage { spanish } [ babel ]
```



```
4 \title{Enumerado con letra}
5 \author{David Pacios}
6 \date{February 2019}
7
8 \begin{document}
9
10 \maketitle
11 \renewcommand{\theenumi}{\Alph{enumi}}
12 \renewcommand{\theenumii}{\alph{enumii}}
13 \begin{enumerate}
14     \item Primer elemento.
15     \item Segundo elemento.
16 \begin{enumerate}
17     \item Tercer elemento.
18     \item Cuarto elemento.
19 \end{enumerate}
20 \end{enumerate}
21 \end{document}
```

### 4.3. Circles (números entre círculos)

Si queremos poner colocar un número entre círculos, sólo tendremos que usar el comando `\circle{}` y poner el número entre corchetes.

. Ejercicio 1 con círculo.

### 4.4. Pie de página bien en un formato bonito con fancyhdr

En el tema anterior vimos cómo poner encabezados y pies de página, pero en este capítulo vamos a ver cómo mejorar esos pies de página.

Primero, vamos a repasar las letras que se colocan detrás de los comandos:

- **E**: Página par.
  - **O**: Página impar.
  - **L**: Parte izquierda.
  - **C**: Parte central.
  - **R**: Parte derecha.
  - **H**: Cabecera.
  - **F**: Pie de página.
-

Una vez hemos visto los comandos que sirven para colocar los encabezados y pies de páginas, vamos a explicar cómo personalizar el pie de página.

Para ello, colocaremos en el preámbulo el paquete `fancyhdr` para instalar el paquete, seguidamente le daremos estilo al pie de página con el comando `\pagestyle{fancy}` y personalizaremos el pie de página con los comandos `\lfoot` y `\cfoot`. Y si además, queremos añadir una línea recta, utilizaremos el comando `\renewcommand{\footrulewidth}{0.4pt}`.

Listing 4.4: Código de pie de página

```

1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage { spanish } [ babel ]
4 %Pie de página personalizado
5 \usepackage { fancyhdr }
6 \pagestyle { fancy }
7 \lhead {}
8 \chead {}
9 \rhead { \bfseries Cabecera parte derecha }
10 \lfoot { by David }
11 \cfoot { OTEA }
12 \rfoot { \thepage }
13 \renewcommand { \headrulewidth } { 0.5 pt }
14 \renewcommand { \footrulewidth } { 0.4 pt }
15 \title { Enumerado con letra }
16 \author { David Pacios }
17 \date { February 2019 }
18 \begin { document }
19 \maketitle
20 \end { document }

```

## 4.5. Matrices y determinantes

Para escribir tanto matrices como determinantes utilizaremos el entorno `array`, que empieza con `\begin{array}{cc}` y termina con `\end{array}`. Donde en el segundo corchete del inicio especificaremos el número de columnas que queremos que tenga la matriz.

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$B = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 & 7 \end{bmatrix}$$

Listing 4.5: Código de matriz y determinante

```

1 \[
2 A=
3 \left( { \begin{array} { cc }

```

```
4     1 & 2 \\
5     3 & 4 \\
6     \end{array} } \right)
7 \\
8 \\
9     B=
10    \left [ {\begin{array}{c}
11      1 & 2 & 3 & 4 & 5 \\
12      3 & 4 & 5 & 6 & 7 \\
13    \end{array} } \right]
14 \\
```

Es muy importante haber puesto en el preámbulo anteriormente el modo matemático y si queremos colocar poner un paréntesis o un corchete ponerlo al principio del comando `\left` y al final del comando `\right`.



# Capítulo 5

# Capítulo 5

## 5.1. Ajedrez

### 5.1.1. Tablero

Las piezas de ajedrez tienen su propio y particular campo de batalla: un tablero de 64 casillas, 32 blancas y 32 negras (a veces se pueden usar otros colores, pero éstos son los más habituales). Lo primero que tenemos que conocer es cómo se coloca dicho tablero. Siempre lo haremos de la misma forma, de tal manera que a nuestra derecha haya una casilla blanca:

Listing 5.1: Código tablero ajedrez

```
1 \documentclass { article }
2 \usepackage [ spanish ] { babel }
3 \usepackage [ utf8 ] { inputenc }
4 \usepackage { skak } % Vamos al jugar al ajedrez
5
6 \title { Ajedrez }
7 \author { David Pacios }
8 \date { February 2019 }
9 \begin { document }
10 \showboard
11 \end { document }
```

El tablero está dividido en filas, columnas y diagonales. Es importante familiarizarse con estos conceptos, ya que serán muy utilizados a lo largo del manual.

El tablero consta de ocho filas, numeradas del 1 al 8, y ocho columnas, definidas por otras tantas letras, de la 'A' a la 'H'. De este modo podemos identificar cada casilla de manera sencilla con sólo dar un número y una letra. Así, por ejemplo, las casillas de las esquinas del tablero serán: a1, h1, a8 y h8.

A continuación vamos a conocer cómo se colocan inicialmente las piezas, el momento en que ambos ejércitos esperan impacientes el comienzo de la batalla. Su distribución es armónica y tiene bastante lógica. Por delante se sitúan los peones, que actúan como

soldados de infantería y protegen al resto de las piezas. El rey y la dama se colocan en las posiciones centrales, ya que son las piezas más importantes y de ese modo están más resguardadas. A izquierda y derecha de los monarcas se colocan el resto de piezas, con las torres situadas en las esquinas del tablero, como si de una fortaleza se tratase. Veamos con nuestros propios ojos donde debe situarse cada pieza:

Listing 5.2: Código empezar partida

```
1 \documentclass { article }
2 \usepackage [ spanish ] { babel }
3 \usepackage [ utf8 ] { inputenc }
4 \usepackage { skak } % Vamos al jugar al ajedre
5
6 \ title { Ajedrez }
7 \ author { David Pacios }
8 \ date { February 2019 }
9 \ begin { document }
10 \ newgame %Comenzar partida
11 \ showboard
12 \ end { document }
```

### 5.1.2. Las piezas

Cada pieza de ajedrez tiene su propia personalidad y significado, podemos decir que incluso tiene vida propia. Sus similitudes con el mundo real son curiosas, lo que nos hace pensar que su creador, allá por el año 600, tenía en mente representar fielmente una batalla.

Sin más, conozcamos en profundidad el movimiento de cada pieza para poder ir pensando en disputar nuestra primera partida:

#### **El rey:**

Sin duda, la pieza más importante. De él depende el resultado final de la partida, detalle que comprobaréis en el siguiente capítulo al conocer el concepto de jaque mate. El monarca reina sobre sus súbditos, los cuales tendrán que velar por su seguridad durante toda la partida. Como hemos dicho, es la pieza más importante, pero no la más poderosa, de hecho sus movimientos son muy limitados. Conozcámoslos.

**CONSEJO:** Trata de mantener a tu rey protegido. De él depende el destino de la partida, así que no facilites la tarea a tu rival dejándolo expuesto (como por ejemplo situándolo en el centro del tablero, lugar donde sería atacado con facilidad).

#### **La dama:**

La dueña y señora del reino, la dama es la pieza más poderosa y decisiva de ambos ejércitos. Por eso debemos cuidarla y moverla con mucho cuidado, ya que su pérdida puede resultar fatal para el resultado final de la partida. ¿Por qué es la pieza más poderosa? Sencillo, es la pieza que tiene mayor alcance ya que se puede desplazar en todas direcciones.

**CONSEJO:** La dama es una pieza vital para poder atacar debido a su gran movilidad, así que no la pongáis en juego demasiado pronto y esperad al momento oportuno para

moverla.

**La torre:**

La torre es una pieza polivalente, puede ser muy útil para la defensa y más tarde convertirse en una poderosa arma ofensiva. Cuando se encuentra en tareas defensivas puede acudir en ayuda de otras piezas que se vean amenazadas. Cuando se usa para atacar forma una pareja temible con su 'hermana', sobre todo cuando ambas se sitúan en la misma fila o columna... ya sabéis, la unión hace la fuerza.

Tras la dama, la torre es la pieza que tiene más movilidad. Se desplaza a lo largo de las columnas y de las filas, siempre en línea recta, lo que le permite moverse por el tablero con gran rapidez.

**El alfil:**

Esta es la pieza que tiene un diseño más extraño. Lo que no todo el mundo sabe es que representa a un miembro del clero, de hecho en inglés se llama bishop, que significa obispo. Su movimiento es muy limitado, ya que siempre debe ir por casillas del mismo color y de forma diagonal.

El alfil resulta muy útil por su largo alcance, ya que puede cruzar el tablero entero en una sola jugada. Cuando quedan pocas piezas en juego su poder aumenta, ya que dispone de mucho espacio por el que moverse.

**El caballo:**

El movimiento del caballo es el más curioso de todas las piezas de ajedrez, ya que lo hace en forma de 'L'. Además, tiene otra característica especial: es la única pieza que puede saltar por encima de las demás, lo que compensa su corto alcance. Estas propiedades le dan una gran movilidad, lo que la convierte en una pieza imprevisible y peligrosa.

Al poder 'saltar' obstáculos, el caballo es la única pieza que se puede poner en juego sin necesidad de mover ningún peón. Como podéis apreciar, se puede desplazar por cualquier rincón del tablero, por lo que siempre trataremos de colocarlo en la mejor posición posible, si es cerca del rey enemigo, mejor.

**El peón:**

Si uno echa un vistazo al tablero con todas sus piezas colocadas, rápidamente pensará que el peón es la más débil de todas... y no estará equivocado. Su movimiento es el más simple: sólo puede avanzar una casilla y siempre, hacia delante. Esta norma sólo se rompe cuando el peón se encuentra en su posición inicial (en la segunda fila), en ese momento podrá mover dos casillas o sólo una, segúnelijamos. Es importante no olvidar que los peones no pueden mover hacia atrás, por lo que debéis meditar a fondo cada movimiento de esta pieza ya que no puede retroceder.

Recuerda, el peón sólo puede mover hacia delante, nunca puede retroceder.

Cuando el peón se encuentra en su casilla inicial podemos escoger entre moverlo una casilla o dos.

Es importante tener en cuenta que las piezas deben ser indicadas en terminología inglesa. Esto es:

- K – Rey (King)
  - Q – Dama (Queen)
  - R – Torre (Rook)
  - N – Caballo (Knight)
  - B – Alfil (Bishop)
-

Más adelante veremos cómo personalizar la entrada para nuestro idioma.

`\mainline{1. e4 e5 2. Nf3 d6}` Asegúrate de que tras el número de cada jugada haya siempre un punto (aunque luego no aparecerá en la salida), de lo contrario no te compilará el código.

Cierra las llaves en el momento que quieras para introducir comentarios o para representar un diagrama con la posición actual empleando `\showboard`

1. e4 e5 2. Nf3 d6

Este es el código completo, incluido el preámbulo, de la transcripción de la partida. Observa como, tras cada interrupción, continuamos la línea principal con sucesivos comandos `\mainline`.

3. d4 Bg4 4.dxe5 Bxf3 5.Qxf3 dxe5 6.Bc4 Nf6 7.Qb3 Qe7 8.Nc3 c6 9.Bg5 b5  
10.Nxb5 cxb5 11.Bxb5+ Nbd7 12. O-O-O  
12... Rd8 13. Rxd7 Rxd7 14. Rd1 Qe6 15. Bxd7+ Nxd7 16. Qb8+ Nxb8 17. Rd8

Presta atención a cómo hemos continuado la partida tras la jugada 12 del blanco. Observa también el uso de la O mayúscula en vez del cero para el enroque, así como el símbolo # para el jaque mate.

## 5.2. Fuentes genéricas

L<sup>A</sup>T<sub>E</sub>X nos permite trabajar con tres familias de fuentes genéricas:

- Con serifas (romanas)
- Sin serifas (sans serif)
- Espaciado fijo (tipo máquina de escribir)

Vamos a explicar un poco en qué consisten.

**¿Qué son las serifas?** Las serifas son esos apéndices decorativos que rematan la escritura de cada letra. Las tienes en todas partes, obsérvalas.

Las serifas mejoran la legibilidad sobre el papel y sobre pantallas con buena resolución, pues ayudan a guiar la vista a través del texto.

A las fuentes que utilizan serifas también se las conoce como romanas.

Por el contrario, un tipo de letra sin serifas carece de esa decoración extra y presenta un aspecto más sobrio. Las fuentes sin serifas suelen emplearse para enfatizar texto, como



contraste a un tipo con serifas. Quedan muy bien para la escritura de títulos. Son muy útiles, además, para la lectura de texto en pantallas con baja resolución: la pixelación estropea las serifas.

**Escritura a máquina** Las fuentes de espaciado fijo tienen la característica, como su nombre indica, de que todas las letras ocupan el mismo tamaño total, tal como hace una máquina de escribir. Los programadores las emplean mucho para escribir código, pues el código fuente suelen presentar muchas indentaciones y les gusta que todo caiga en su sitio y el programa quede legible.

**Computer Modern** LaTeX tiene una familia de fuentes por defecto para cada uno de estos tres tipos:

- Computer Modern Roman (con serifas)
- Computer Modern Sans Serif (sin serifas)
- Computer Modern Typewriter (espaciado fijo)

Ya te has dado cuenta de que, por defecto, LaTeX no utiliza Times (que es una mierda), sino Computer Modern, una familia de fuentes que, además, posee variantes para los tres tipos.

Puedes cambiar el tipo base por defecto, si lo deseas. Deberás facilitar una familia adecuada para cada uno de los tres tipos, o elegir un paquete adecuado que lo gestione por ti.

Por el momento quédate con la idea de que dispones de tres familias básicas que puedes emplear para enfatizar.

**Los comandos** Los comandos para cambiar la familia de la fuente son:

```
\textrm (fuente romana, con serifas)
\textsf (fuente sin serifas)
\texttt (fuente de espaciado fijo)
```

Recuerda que cuando utilizabas comandos, el texto afectado se facilitaba como parámetro entre las llaves. Por ejemplo:

La gente que no viene  
escribiría la frase sin serifas.

---

**Las declaraciones** Las declaraciones eran comandos que actuaban como conmutadores. A partir de ellas se activaba un determinado modo de funcionamiento que desaparecía cuando finalizaba el ámbito de actuación u otra declaración activaba un modo diferente incompatible con el anterior.

Las declaraciones para cambiar la familia de la fuente son:

`\rmfamily` (fuente romana, con serifas) `\sffamily` (fuente sin serifas) `\ttfamily` (fuente de espaciado fijo)

### 5.3. Indentación de párrafos

En  $\LaTeX$  por defecto, indenta la primera línea de cada párrafo. De este modo, salta a la vista donde empieza uno y termina otro, incluso sin existir separación entre ellos (algo que, como vimos, también sucede por defecto).

Naturalmente, este comportamiento puede ser modificado fácilmente y podemos especificar con precisión cuánto queremos que quede indentada la primera línea de cada párrafo.

Es una cuestión tan simple como modificar una de las magnitudes de longitud de  $\LaTeX$ . Es semejante a lo que hicimos con `\parskip` para controlar el espaciado entre párrafos.

Deja que te presente la magnitud `\parindent`.

Imagina que quieres definir una indentación de 12pt en la primera línea de todos los párrafos del documento. Escribe el siguiente comando en el preámbulo: `\setlength{\parindent}{12pt}`

Es exactamente lo mismo que hacíamos con `\parskip`: el comando `\setlength`, seguido de la magnitud que queremos modificar, entre llaves, y de su valor, entre llaves también, sin olvidar especificar las unidades. El comando tiene dos pares de llaves porque son dos parámetros obligatorios. Recuerda que los parámetros opcionales figuraban entre corchetes.

El preámbulo es, desde luego, el mejor sitio para este tipo de comandos que afectan al diseño global, pero también puedes hacer uso de ellos dentro del cuerpo del documento. En este caso, los cambios en la sangría sólo aparecerán en los párrafos posteriores a su aparición y se mantendrán vigentes hasta que otro comando similar vuelva a alterar su valor.

Si deseas limitar su efecto a uno o más párrafos, puedes incluir el comando dentro de un grupo delimitado por llaves. Es decir:

`{\setlength{\parindent}{12pt} ... párrafos afectados....}`

Sólo aquellos párrafos contenidos entre las llaves se verán afectados; los restantes mantendrán la indentación especificada en el preámbulo (o la que hay por defecto, de no indicar una).

Las buenas prácticas recomiendan emplear unidades de longitud relativas en vez de absolutas. Recuerda, por ejemplo, la `em`, que equivale a la anchura de la letra `m`. Esta anchura es, obviamente, mayor, si el tamaño de la fuente es mayor. Hacer esto garantiza que la indentación sea proporcional al tamaño de la fuente. Si en algún momento quieres modificar el tamaño base en tu documento, las sangrías se ajustarán proporcionalmente, algo que te evitará tener que retocarlas después.

El siguiente comando establece la indentación a una vez y media la anchura de una letra `eme`:

```
\setlength{\parindent}{1.5em}
```

La indentación de los párrafos es algo común en el mundo anglosajón, pero no tanto en el nuestro, pues preferimos no hacerlo.

El siguiente comando, en el preámbulo del documento, anula el sangrado de los párrafos:

```
\setlength{\parindent}{0cm}
```

Fíjate en que, a pesar de que el valor es cero, se necesita indicar las unidades. Habría sido lo mismo, desde luego, escribir `0pt`, `0mm` o `0em`.

Puede que desees que sólo un párrafo concreto quede sin indentación. Podrías utilizar el comando anterior encerrado en un grupo con llaves, aunque mucho más sencillo es hacer uso de otro comando simple:

```
\noindent
```

Coloca este comando al principio del párrafo que no desees sangrar y desactivarás momentáneamente, sólo durante ese párrafo, la indentación existente.

Si no te gusta indentar los párrafos, asegúrate de que te guste agregar una separación entre ellos tocando la magnitud `\parskip`. De no hacerlo, si la última línea de un párrafo es larga, no podrás distinguir si estás o no en el mismo párrafo o en el siguiente.

O, mejor aún, utiliza el paquete `parskip` (el mismo nombre que la magnitud), que ya tratamos, y que se ocupa de establecer a cero el sangrado y de definir una línea de separación entre párrafos. Además, mejora el tratamiento que se le da a las listas, que pueden verse más afectadas por haber tocado la magnitud `\parskip`.

---

## 5.4. Espacio entre párrafos

Hay una magnitud que controla la separación de líneas en  $\text{\LaTeX}$ :  $\backslash\text{baselineskip}$ .

Puede ser tentador, entonces, introducir un simple comando para modificar esa magnitud, como

```
 $\backslash\text{setlength}\{\backslash\text{baselineskip}\}\{18\text{pt}\}$ 
```

Sin embargo, eso NO FUNCIONA.

La cuestión es que  $\text{\LaTeX}$  sobrescribe  $\backslash\text{baselineskip}$  y lo ajusta dinámicamente en función de aspectos como, por ejemplo, el tamaño de la fuente.

Típicamente, para una fuente de tamaño normal de 10pt, la separación entre líneas es 12pt. Pero, si esa misma fuente la empleas en versión Huge, puedes encontrarte con una separación de hasta 30pt.

Para modificar el interlineado recurrimos a la macro  $\backslash\text{baselinestretch}$ , que no es más que un multiplicador del valor de  $\backslash\text{baselineskip}$ . Por defecto, su valor es 1.

Si hacemos que el valor del multiplicador sea 2, conseguiremos que la separación entre líneas sea del doble. El comando siguiente, que introduciremos en el preámbulo, redefina la macro  $\backslash\text{baselinestrech}$ :

```
 $\backslash\text{renewcommand}\{\backslash\text{baselinestretch}\}\{2\}$ 
```

Si en vez del doble deseáramos una distancia y media:

```
 $\backslash\text{renewcommand}\{\backslash\text{baselinestretch}\}\{1.5\}$ 
```

De este modo, podemos olvidarnos tranquilamente del ajuste dinámico que  $\text{\LaTeX}$  realiza sobre  $\backslash\text{baselineskip}$ . Si ante una fuente normal separaba las líneas 12pt y 30pt en versión Huge, con un multiplicador de 1.5 las separará 18pt y 45pt, respectivamente.

## 5.5. Inserción de documentos PDF

Para poder agregar otros archivos PDF a nuestro documento necesitamos hacer uso del paquete `pdfpages`:

```
 $\backslash\text{usepackage}\{\text{pdfpages}\}$ 
```

Los PDF que insertemos no tienen que ser necesariamente generados en  $\text{\LaTeX}$ ; todos sirven.

En el punto del documento en el que deseemos que se realice la inserción utilizamos el comando:

```
\includepdf[pages inicial-final]{nombre_del_documento}
```

Si el punto de inserción está en mitad de una página, L<sup>A</sup>T<sub>E</sub>X forzará un salto de página llegado ese punto para que el PDF se agregue al comienzo de una nueva.

Observa que el primer parámetro es opcional (escrito entre corchetes). Si lo omites sólo se agregará la primera página del PDF.

El rango de páginas se indica mediante dos cifras separadas por un guión.

Por ejemplo:

**[pages=3-7]** Agrega sólo las páginas comprendidas entre la 3 y 7, ambas incluidas.

Si no especificamos un valor inicial L<sup>A</sup>T<sub>E</sub>X asumirá que quieres insertar desde el principio:

**[pages=-10]** Agrega desde el comienzo hasta la página 10.

Del mismo modo, si no indicamos un valor final sobreentenderá que es hasta el final del documento:

**[pages=5-]** Agrega desde la página 5 hasta la última.

Si queremos que se inserte el documento completo, omitimos los valores inicial y final, pero aún así debemos incluir el guión:

**[pages=-]** Agrega el documento completo, de principio a fin.

El parámetro obligatorio, el nombre del fichero, se indica entre llaves. Puedes omitir, si lo deseas, la extensión, pues se da por supuesto que estamos lidiando con PDFs.

Eso sí, si el PDF no está en el mismo directorio que el .tex asegúrate de incluir también la ruta (absoluta o relativa). Presta mucha atención y, aunque estés trabajando en una máquina Windows, usa como separador de directorio la barra al estilo Unix, /, y no \ pues esta última es un símbolo reservado de L<sup>A</sup>T<sub>E</sub>X (el indicador de comando).

Veamos algunos ejemplos:

```
\includepdf{comparativa}
```

Agrega la primera página del fichero comparativa.pdf.

```
\includepdf[pages25-27]comparativa=
```

Inserta las páginas 25, 26 y 27 del fichero comparativa.pdf.

---

```
\includepdf[pages-]running/clasificacion=
```

Inserta la totalidad del documento clasificacion.pdf, almacenado en el directorio running, presente en la misma carpeta que el fichero .tex.

# Capítulo 6

# Capítulo 6

## 6.1. ¿Qué es Beamer?

Beamer es una clase de documento que está diseñado para realizar presentaciones. Con este sistema, podemos colocar fórmulas y cajas con comodidad. Para poder utilizarlo, definiremos en el preámbulo la clase `beamer`. Y utilizaremos el entorno `frame` para dar el nombre a cada diapositiva.



Figura 6.1: Beamer básico

Listing 6.1: Código Beamer básico

```
1 \documentclass {beamer}
2 \usepackage [ utf8 ] {inputenc}
3 \usepackage {spanish} [ babel ]
4
5 \title {Beamer básico}
6 \author {David Pacios}
7 \date {February 2019}
8
9 \begin {document}
10 \frame { \titlepage }
11 \input {Frame1.tex}
12 \end {document}
```

En este código podemos ver un nuevo comando, el comando `\input{}`, y entre corchetes, colocaremos la situación de lo que queramos insertar en el documento.

Y finalmente, tenemos otro comando para insertar las diapositivas, que es el comando `\include{}` que funciona igual que el anteriormente descrito.

Listing 6.2: Código Beamer básico con include

```
1 \documentclass {beamer}
2 \usepackage [ utf8 ] {inputenc}
3 \usepackage {spanish} [ babel ]
4
5 \title {Beamer básico}
6 \author {David Pacios}
7 \date {February 2019}
8
9 \begin {document}
10 %añadir los Frames con \include
11 \include {Frame/Frame1} %Título
12 \include {Frame/Frame2}
13
14 \end {document}
```

## 6.2. Creación de bloques

Para crear bloques dentro del documento utilizaremos el entorno `block`, que comienza con `\begin{block}{}{}` y entre corchetes colocamos el título que le queramos dar el bloque, y finalizamos el entorno con `\end{block}`.



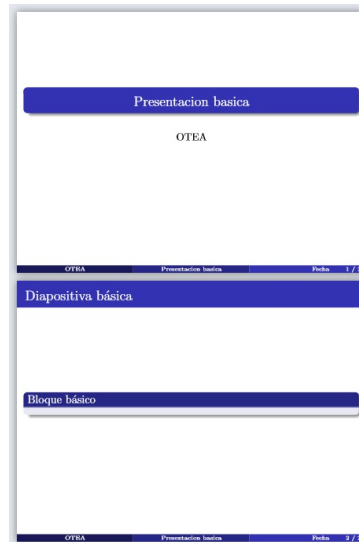


Figura 6.2: Bloque básico

Listing 6.3: Código Beamer bloque básico

```

1 \begin{frame}{Diapositiva básica}
2 \begin{block}{Bloque básico}
3 \end{block}
4 \end{frame}

```

Además de poder colocar bloques normales, podemos utilizar bloques de alerta con el comando `alertblock`.



Figura 6.3: Bloque alerta básico

Listing 6.4: Código Beamer bloque alerta básico

```

1 \begin{frame}{Diapositiva básica}

```

```
2 \begin{alertblock}{Bloque alerta básico}
3 Texto bloque
4 \end{alertblock}
5 \end{frame}
```

También tenemos otro tipo de bloque básico, que es el comando `exampleblock` para poner recuadros de ejemplo.

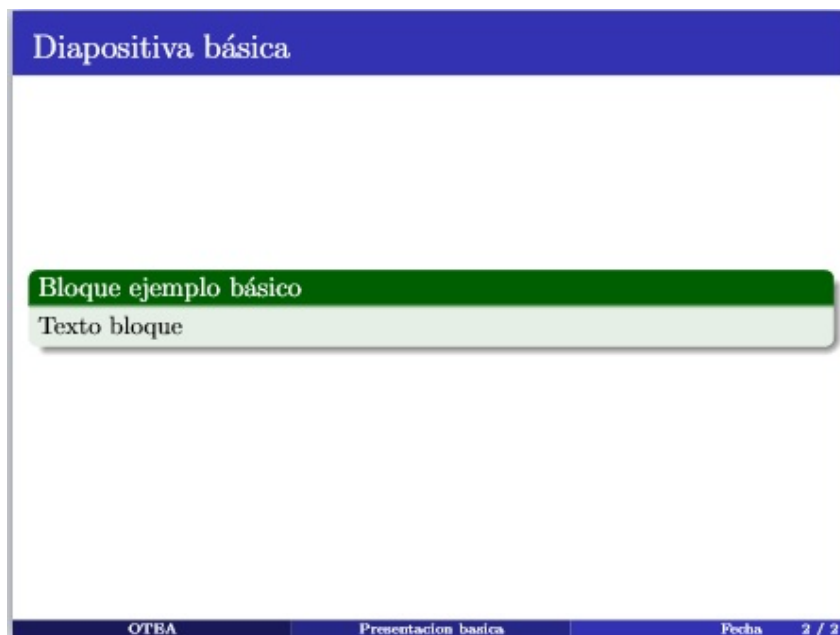


Figura 6.4: Bloque ejemplo básico

Listing 6.5: Código Beamer bloque ejemplo básico

```
1 \begin{frame}{Diapositiva básica}
2 \begin{exampleblock}{Bloque ejemplo básico}
3 Texto bloque
4 \end{exampleblock}
5 \end{frame}
```

### 6.2.1. Cajas con color

Para crear cajas con color vamos a utilizar el entorno `beamercolorbox`, y para definir el color utilizaremos el comando `\setbeamercolor{nombre}{definición color}` en el preámbulo.

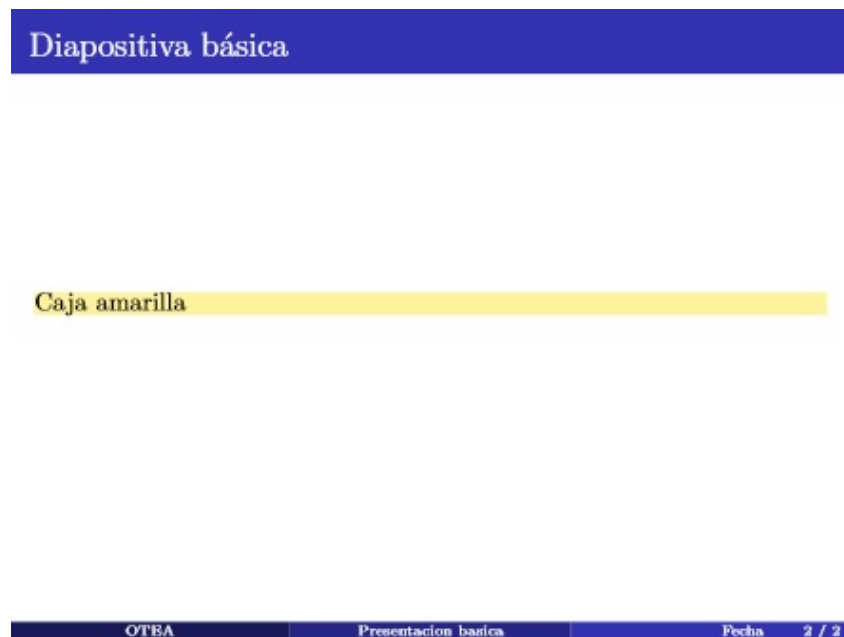


Figura 6.5: Bloque color amarillo

Listing 6.6: Código Beamer bloque amarillo

```

1 %Preambulo main
2 \documentclass {beamer}
3 \setbeamercolor {postit} {bg=yellow!50!white}
4 \begin{document}
5 \include {Frame/Frame1} %Titulo
6 \include {Frame/Frame2}
7 \end{document}
8 %En la diapositiva
9 \begin{frame} {Diapositiva básica}
10 \begin{beamercolorbox} {postit}
11 Caja amarilla
12 \end{beamercolorbox}
13 \end{frame}

```

También tenemos una serie de opciones de alineamiento para las cajas coloreadas que son las siguientes:

- Dimensiones de la caja:
  - Anchura: `wd`.
  - Profundidad: `dp`.
  - Altura: `ht`.
- Alineación del contenido en la caja:
  - `left`.
  - `right`.

- center.
- Separación entre texto y marco: `\sepDistancia`.

## 6.3. Animaciones

Finalmente, en nuestras presentaciones podemos introducir unas animaciones, para que nos vayan saliendo las distintas opciones de distintas formas en la diapositivas. Son más complicadas de programar y se van a explicar detenidamente cada una de ellas.

### 6.3.1. Generalidad

El primer comando que vamos a explicar es el comando `\pause` para colocar una pausa.



Figura 6.6: Comando pause

Listing 6.7: Código comando pause

```

1 \begin{frame}{Diapositiva básica}
2 \begin{beamercolorbox}{postit}
3 \begin{itemize}
4     \item Primer comando \pause
5     \item Segundo comando \pause
6     \item Tercer comando
7 \end{itemize}

```



Figura 6.7: Comando pause

Listing 6.8: Código comando pause

```
1 \begin{frame}{Diapositiva básica}
2 \begin{beamercolorbox}{postit}
3 \begin{itemize}
4     \item Primer comando \pause
5     \item Segundo comando \pause
6     \item Tercer comando
7 \end{itemize}
8 \end{beamercolorbox}
9 \end{frame}
```

### 6.3.2. Especificaciones

Hay algunos otros comandos que admiten unas especificaciones que están en el comando `<Rango>` en el que ponemos colocar lo siguiente:

- 1- → Del primero en adelante.
- -3 → Hasta el tercer paso.
- 2-5 → Del segundo al quinto.
- 1-3, 5 → Del primero al tercero, incluyendo el quinto.

El comando que colocaremos será `\Comando<Rango>\{Contenido\}`.

### 6.3.3. Comando onslide

Con este comando, seleccionaremos el orden en que queramos que aparezca nuestro contenido. Y lo colocaremos `\onslide<Rango>\{Texto\}` así.

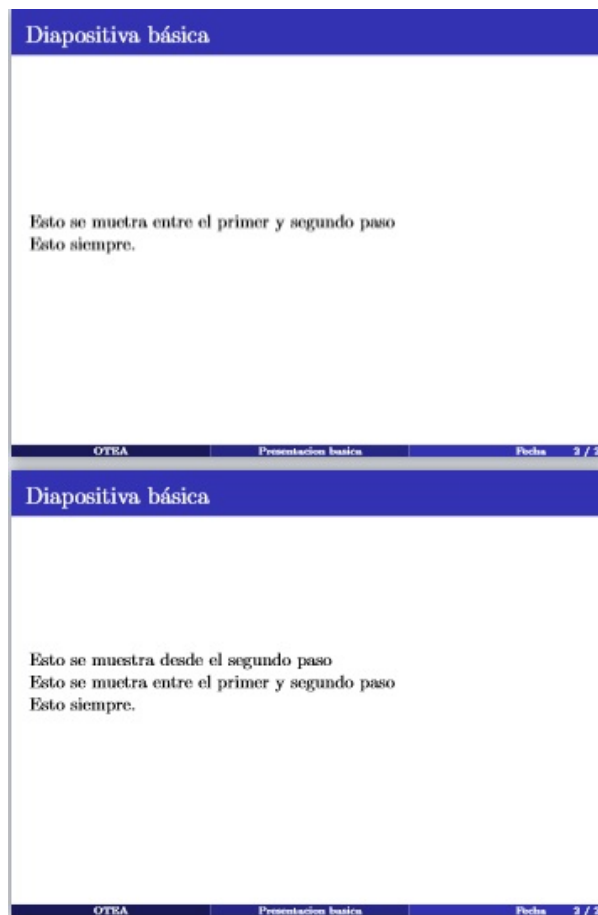


Figura 6.8: Comando onslide

Listing 6.9: Código comando onslide

```

1 \begin{frame}{Diapositiva básica}
2 \onslide<2->{Esto se muestra desde el segundo paso}\\
3 \onslide<1-2>{Esto se muestra entre el primer y segundo paso}\\
4 Esto siempre.
5 \end{frame}

```

### 6.3.4. Comando only

Este comando es similar al anterior, pero las sustituciones a la siguiente diapositiva se realizan de forma animada.

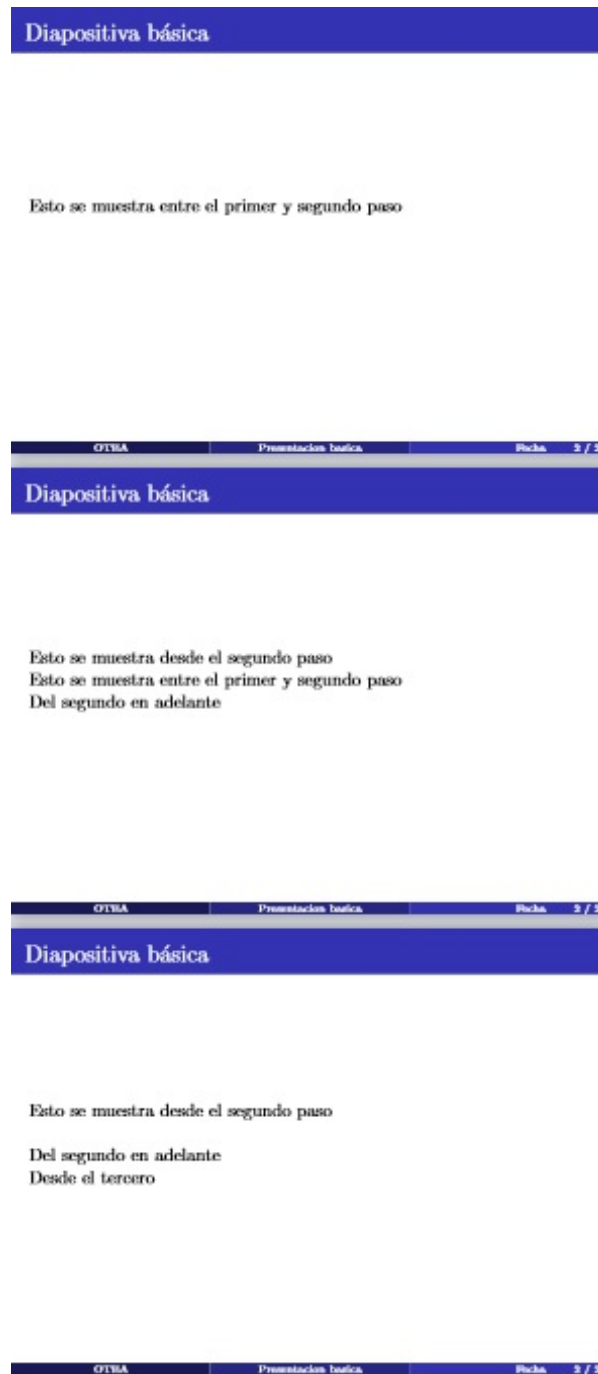


Figura 6.9: Comando only

Listing 6.10: Código comando only

```

1 \begin{frame}{Diapositiva básica}
2 \only<2->{Esto se muestra desde el segundo paso}\\
3 \onslide<1-2>{Esto se muestra entre el primer y segundo paso}\\
4 \onslide<2->{Del segundo en adelante}\\
5 \only<3>{Desde el tercero}
6 \end{frame}

```



## 6.4. Efectos de transparencia

El comando `\setbeamcovered{Opcion}` nos permite colocar el texto no visualizado mostrado en las animaciones. Antes de poder utilizarlo, colocaremos en el preámbulo el comando `\mode<presentation>{}` con las siguientes opciones:

- **invisible**: Texto oculto totalmente.
- **dynamic**: Varia grado de visibilidad.



Figura 6.10: Comando invisible

Listing 6.11: Código comando only

```

1 %Preámbulo
2 \documentclass {beamer}
3 \setbeamercolor {postit} {bg=yellow!50!white}
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Modelo de presentación
5 \mode<presentation>{
6   \usetheme{Madrid}           %Estilo Madrid (Muy bonito)
7   \usecolortheme{default}    %Colores bases
8   \usefonttheme{serif}       %Tipo de fuente, recomendable serif
9   \setbeamertemplate{navigation symbols}{}
10  \setbeamertemplate{caption}[numbered]
11  \setbeamercovered{invisible}
12 }
13 %Diapositiva
14 \begin{frame}{Diapositiva básica}
15 \onslide<1->{Esto se muestra desde el segundo paso}\\
16 \onslide<2-3>{Esto se muestra entre el primer y segundo paso}\\
17 Esto siempre.
18 \end{frame}

```

## 6.5. Comando visible

Cuando no funcione el comando anterior, podremos utilizar el comando `\visible<Rango>{Texto}` para la misma función.

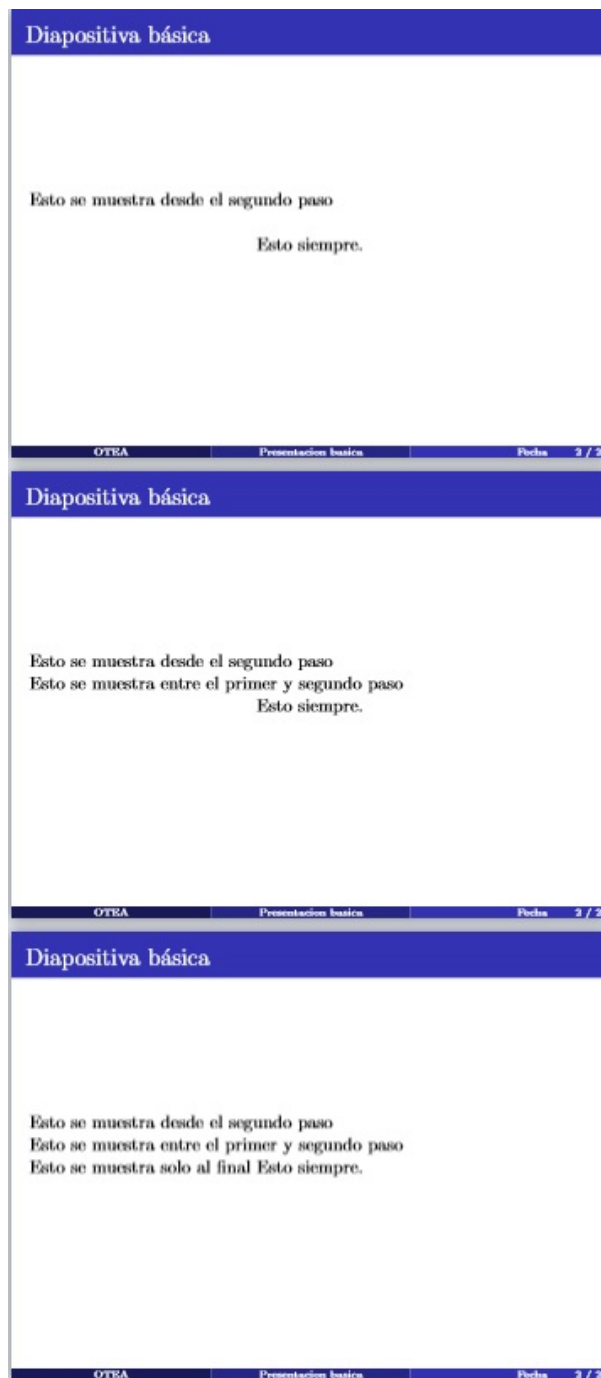


Figura 6.11: Comando visible

Listing 6.12: Código comando visible

```

1 \begin{frame}{Diapositiva básica}
2 \onslide<1->{Esto se muestra desde el segundo paso}\\
3 \visible<2-3>{Esto se muestra entre el primer y segundo paso}\\
4 \onslide<3>{Esto se muestra solo al final}
5 Esto siempre.
6 \end{frame}

```

### 6.5.1. Estilos de la presentación

Los estilos de la presentación se encuentran en el preámbulo y para ello, se utiliza el comando `\usetheme{}`. Entre los distintos temas podemos encontrar los siguientes:

- Temas sin barra de navegación:
  - default.
  - Bergen.
  - Boadilla.
  - Madrid.
  - AnnArbor.
  - CambridgeUS.
  - Pittsburgh.
  - Rochester.
- Temas con barrade navegación tipo árbol:
  - Antibes.
  - JuanLesPins.
- Temas con tabla de contenido lateral:
  - Berkeley.
  - PaloAlto.
  - Goettingen.
  - Marburg.
  - Hannover.
- Temas con navegación en miniframe:
  - Berlin.
  - Ilmenau.
  - Dresden.
  - Darmstadt.
  - Frankfurt.
  - Singapore.
  - Szeged.
- Temas con menús de sección y subsección:
  - Copenhagen.
  - Lueback.
  - Malmoe.
  - Warsaw.



```
8 \usefonttheme{serif} %Tipo de fuente, recomendable serif
9 \setbeamertheme{navigation symbols}{}
10 \setbeamertheme{caption}[numbered]
11 \setbeamercovered{invisible}
12 }
```

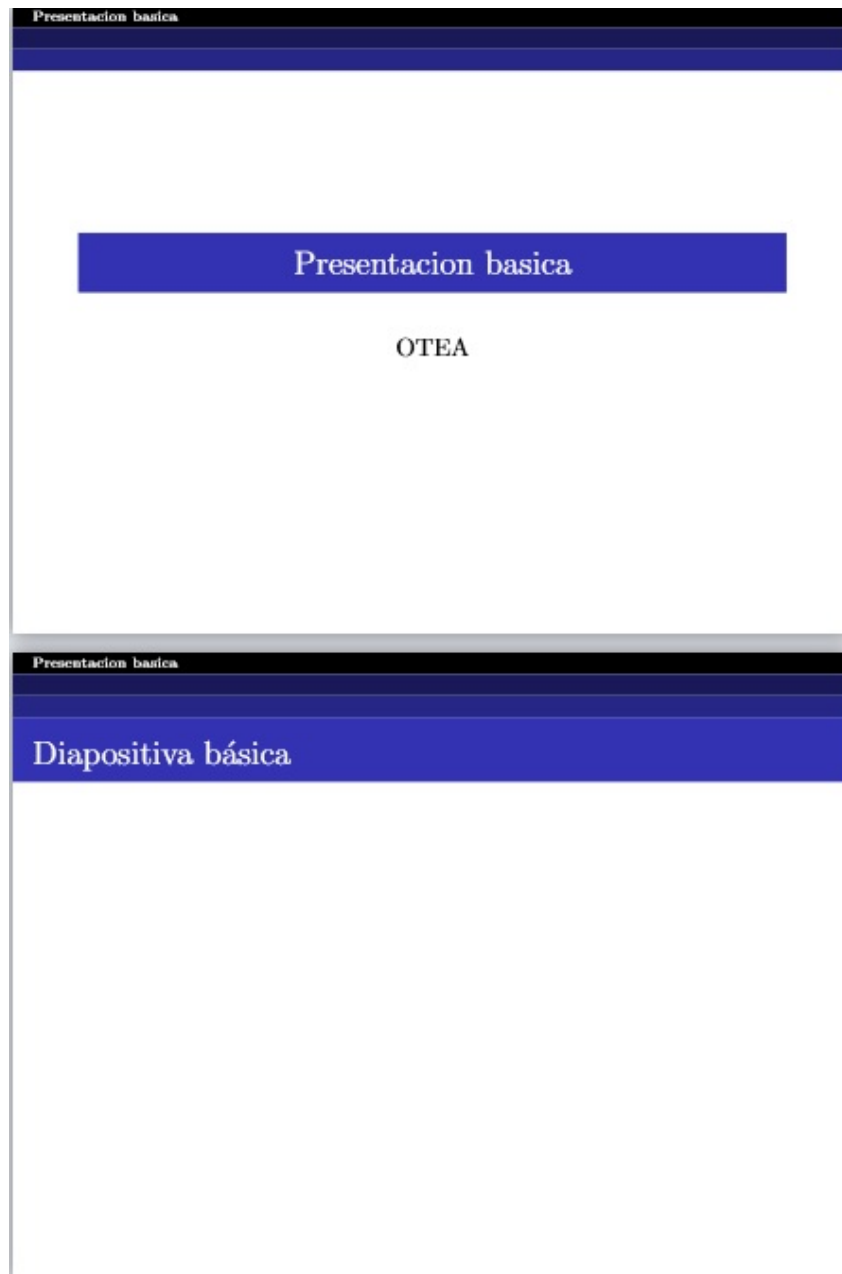


Figura 6.13: Presentación tipo Antibes

Listing 6.14: Código comando estilo Antibe

```

1 \documentclass {beamer}
2 \newcommand{\pascal}{\LARGE$\{\displaystyle \mathbf {P\!\!\!^{\{\}_-
   \scriptstyle A}\!\!\!\!\!\;\;\!S\!\_ {\displaystyle C}\!\!A\!\!\!^
   \{\}_ {\scriptstyle L}\}\}$}
3 \setbeamercolor {postit}{bg=yellow!50!white}
4 %%%%%%%%%%%%%%% Modelo de presentación
5 \mode<presentation>{
6   \usetheme{Antibes}
7   \usecolortheme{default} %Colores bases
8   \usefonttheme{serif} %Tipo de fuente, recomendable serif
9   \setbeamertheme{navigation symbols}{}
10  \setbeamertheme{caption}[numbered]
11  \setbeamercovered{invisible}
12 }
    
```

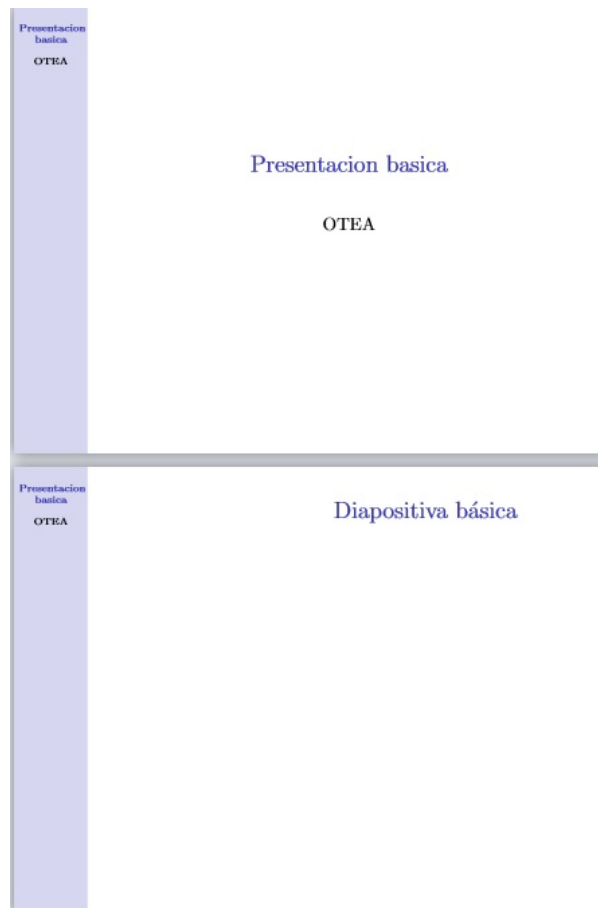


Figura 6.14: Presentación tipo Hannover

Listing 6.15: Código comando estilo Hannover

```

1 \documentclass {beamer}
2 \newcommand{\pascal}{\LARGE${\displaystyle \mathbf {P\!\!\!^{\!}_{\!}}_
   {\scriptstyle A}}\!\!\!\!\!\!\!;\!;\!S\!\!_{\displaystyle C}\!\!A\!\!\!^
   {\!}_{\scriptstyle L}}}$}
3 \setbeamercolor {postit} {bg=yellow!50!white}
4 %%%%%%%%%%%%%%%Modo de presentación
5 \mode<presentation>{
6   \usetheme{Hannover}
7   \usecolortheme{default} %Colores bases
8   \usefonttheme{serif} %Tipo de fuente, recomendable serif
9   \setbeamertemplate{navigation symbols}{}
10  \setbeamertemplate{caption}[numbered]
11  \setbeamercovered{invisible}
12 }

```

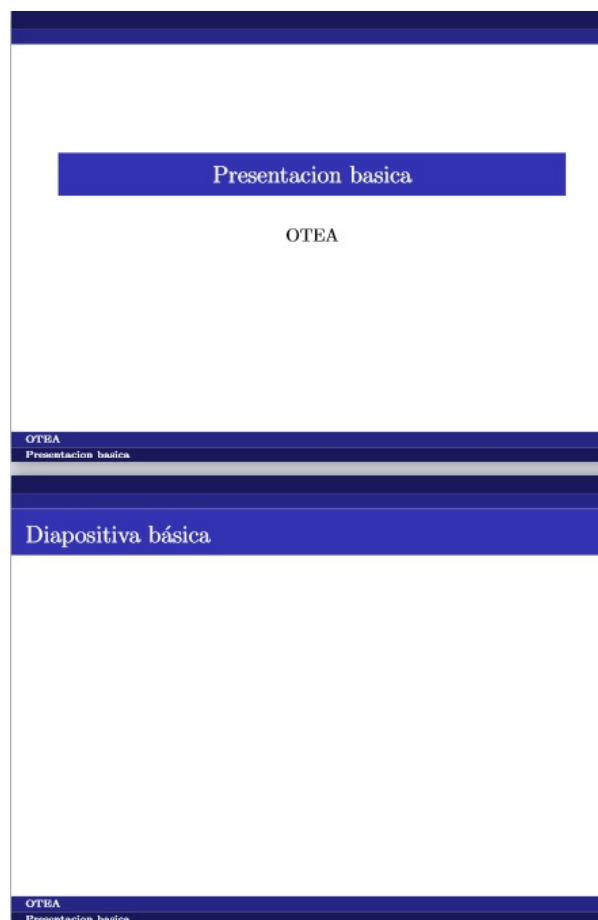


Figura 6.15: Presentación tipo Berlin

Listing 6.16: Código comando estilo Berlin

```

1 \documentclass {beamer}

```



```

2 \newcommand{\pascal}{\LARGE$\displaystyle \mathbf {P\!\!\!^{\!_
   \scriptstyle A}}\!\!\!\!\!\;\!S\!\_ {\displaystyle C}\!\!A\!\!\!^
   \!_ {\scriptstyle L}}}$
3 \setbeamercolor {postit}{bg=yellow!50!white}
4 %%%%%%%%%%%%%%% Malmoe de presentación
5 \mode<presentation>{
6   \usetheme{Berlin}
7   \usecolortheme{default} %Colores bases
8   \usefonttheme{serif} %Tipo de fuente, recomendable serif
9   \setbeamertemplate{navigation symbols}{}
10  \setbeamertemplate{caption}[numbered]
11  \setbeamercovered{invisible}
12 }

```

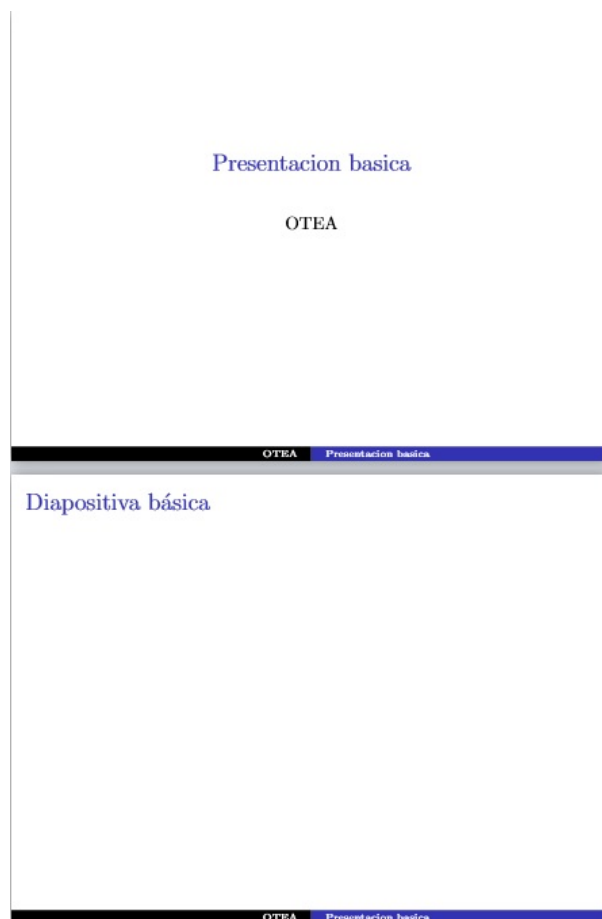


Figura 6.16: Presentación tipo Malmoe

Listing 6.17: Código comando estilo Malmoe

```

1 \documentclass {beamer}

```

```
2 \newcommand{\pascal}{\LARGE${\displaystyle \mathbf {P\!\!\!^{\{}_
  {\scriptstyle A}}\!\!\!\!\!\!\\;S\!_ {\displaystyle
3 C}\!\!A\!\!\!^{\{}_ {\scriptstyle L}}}\!$}
4 \setbeamercolor {postit}{bg=yellow!50!white}
5 %%%%%%%%%%%%%%% Modo de presentación
6 \mode<presentation>{
7   \usetheme{Malmoe}
8   \usecolortheme{default} %Colores bases
9   \usefonttheme{serif} %Tipo de fuente, recomendable serif
10  \setbeamertemplate{navigation symbols}{}
11  \setbeamertemplate{caption}[numbered]
12  \setbeamercovered{invisible}
13 }
```

Además de poder cambiar el estilo de la diapositiva, podremos cambiar el estilo de los bloques, entornos e itemizados con el comando `\useinnertheme{Nombre del tema}`, los diseños del color con el comando `\usecolortheme{Nombre del tema}` y la apariencia de la fuente con el comando `\usefonttheme{Nombre del tema}`.

Entre los distintos tipos de diseños de bloques, entornos e itemizados podemos encontrar:

- default.
- circles.
- rectangles.
- rounded.
- inmargin.

También podemos cambiar el diseño del color con los siguientes diseños:

- default.
- structure.
- sidebartab.
- albatross.
- beetle.
- crane.
- dove.
- fly.
- Y muchos más estilos.

Y finalmente, podremos cambiar los tipos de letra con los siguientes estilos:

- serif.

- `structurebold`.
  - `structureitalicserif`.
  - `structuresmallcapserif`.
-



# Capítulo 7

# Capítulo 7

## 7.1. Diagrama de barras, círculos

### 7.1.1. Diagrama de barras

Para realizar un diagrama de barras vamos a instalar los paquetes `pgfplots` para habilitar la gráfica y `textcomp`, para añadirle un texto.

Una vez se haya instalado, comenzaremos con el entorno `tikzpicture`, para ello empezaremos con el comando `\begin{tikzpicture}` y terminaremos con el comando `\end{tikzpicture}`, pero no sólo estos comandos nos serán útiles para realizar la gráfica de barras, sino que deberemos añadir el entorno `axis`, este entorno comenzará con el comando `\begin{axis}` y terminará con el comando `\end{axis}`. Con este último entorno seremos capaces de añadir los datos de nuestra gráfica de barras y la leyenda. En este tipo de gráficos encontraremos los siguientes nuevos comandos:

- `x tick label style{/pgf/number format/1000 sep=}`: Este comando define el estilo del diagrama. Además, podremos incluir los ejes con el comando `\addplot` y si queremos añadir los valores del eje  $y$  con el comando `\ybar`.
- `enlargelimits0.05`: Alarga los límites de la barra tanto se indique. En este caso, se le daría un límite de altura a la barra de 0.05.
- `legend style{at={(0.5,-0.2)}, anchor=north, legend columns=-1}`: Indica la posición y el ancho de la leyenda, en este caso, sería muy cercana al eje  $x$ .
- `ybar interval0.7`: Indica la delgadez de cada barra, si se indica 1 significa que las barras estarán una al lado de la otra y 0 significa que no habrá barras, sino líneas verticales.

A continuación, se mostrará un diagrama de barras con su código de ejemplo:

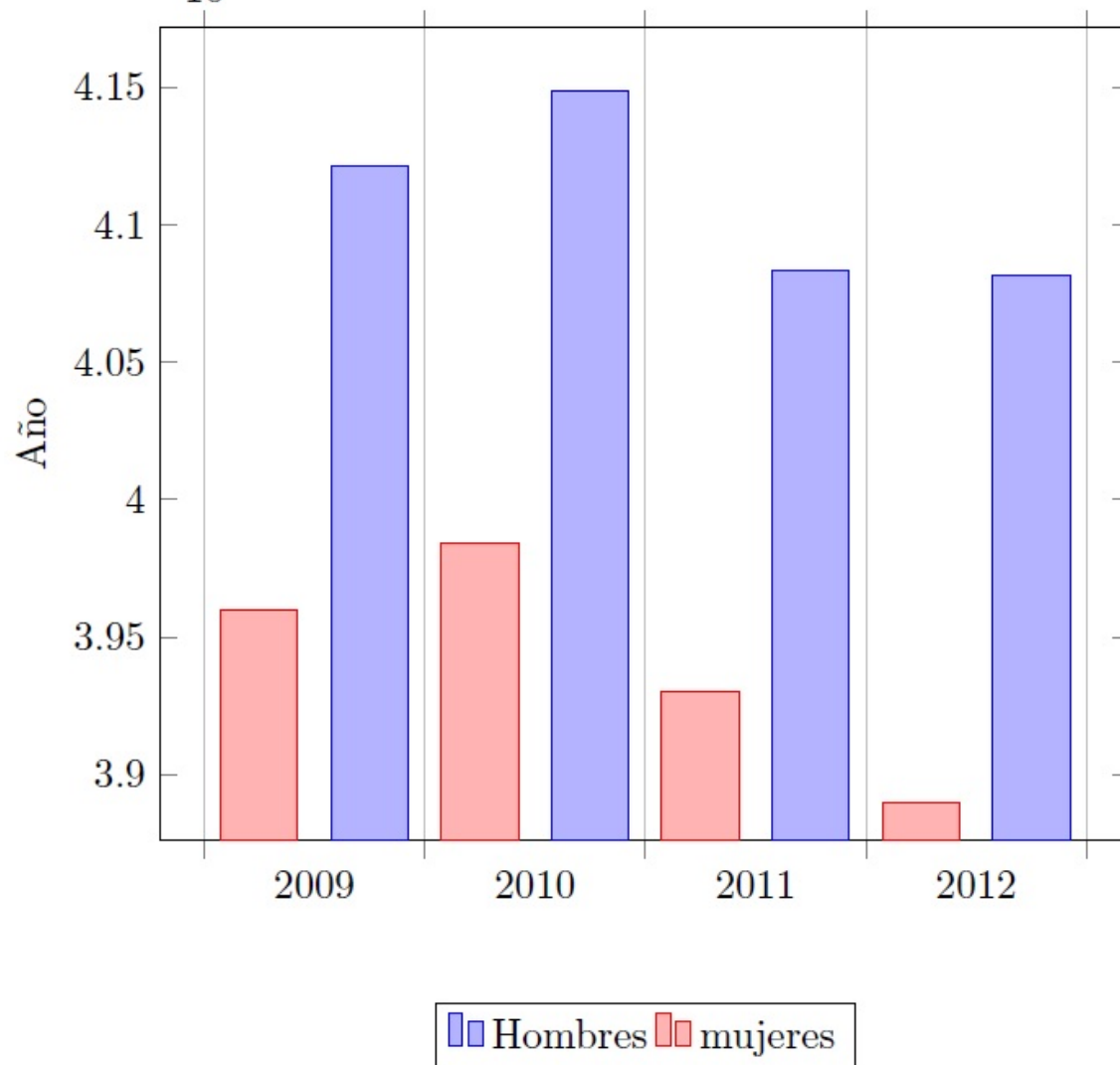
Gráfica de barras:  
·10<sup>5</sup>

Figura 7.1: Gráfica de barras

Listing 7.1: Código diagrama básico

```

1 \documentclass { article }
2 \usepackage [ margin=0.5in ] { geometry }
3 \usepackage [ utf8 ] { inputenc }
4 \usepackage { textcomp }
5 \usepackage { pgfplots }
6 \pgfplotsset { width=10cm, compat=1.9 }
7
8
9 \begin { document }
10
11 Gráfica de barras :

```

```

18 \begin{tikzpicture}
19 \begin{axis}[
20     x tick label style={
21         /pgf/number format/1000 sep=},
22     ylabel=Año,
23     enlargelimits=0.05,
24     legend style={at={(0.5,-0.2)},
25         anchor=north,legend columns=-1},
26     ybar interval=.7,
27 ]
28 \addplot
29     coordinates {(2012,408184) (2011,408348)
30         (2010,414870) (2009,412156) (2008,415 838)};
31 \addplot
32     coordinates {(2012,388950) (2011,393007)
33         (2010,398449) (2009,395972) (2008,398866)};
34 \legend{Hombres, mujeres}
35 \end{axis}
36 \end{tikzpicture}
37 \end{document}

```

### 7.1.2. Diagrama de círculos

Para realizar un diagrama de círculos vamos a utilizar el paquete `pgf-pie`.

El entorno del diagrama de círculos comienza con `\begin{tikzpicture}` y termina con `\end{tikzpicture}`, y dentro del entorno con el comando

`\pie[Dimensiones del círculo]{Contenido del círculo}`. Para realizar cada porcentaje utilizaremos `/` y separemos cada porcentaje por comas. Para tenerlo más claro podremos un ejemplo:

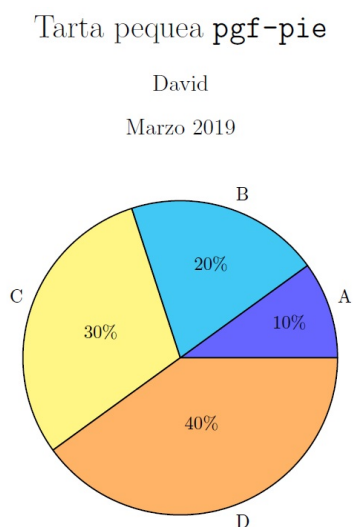


Figura 7.2: Diagrama de círculos

Listing 7.2: Código diagrama círculos

```

1 \usepackage{pgf-pie}
2
3 \title{Tarta pequeña \texttt{pgf-pie}}
4 \author{David}
5 \date{Marzo 2019}
6
7 \begin{document}
8 \maketitle
9
10 \centering
11
12 \begin{tikzpicture}
13 \pie{10/A, 20/B, 30/C, 40/D}
14 \end{tikzpicture}
15 \end{document}

```

## 7.2. Esquema con llaves

Para realizar un esquema con llaves necesitaremos instalar en el preámbulo el paquete [schemata](#).

Si queremos realizar una llave simple utilizaremos el comando `\schema{\schemabox{Contenido a la izquierda de la llave}}{\schemabox{Contenido a la derecha}}`. Para tenerlo más claro vamos a poner un ejemplo:

Contenido 1Contenido 2

Listing 7.3: Código esquema simple

```

1 \usepackage{schemata}
2 \schema{\schemabox{Contenido 1}}{\schemabox{Contenido 2}}

```

Con este comando algo enrevesado se puede crear un diagrama simple, pero renombrando con el comando `\newcommand\diagram[2]{\schema{\schemabox{#1}}{\schemabox{#2}}}` hace que podamos realizar un diagrama más sencillo. Y podremos realizar el siguiente esquema:

**Criterios de exigencia** - Constancia: **Trabajar mucho**

- Tiempo: **El tiempo empleado**

- Otros- Buena presentación

- ...

Listing 7.4: Código esquema de llaves complejo

```

1 \usepackage{schemata}
2 \newcommand\diagram[2]{\schema{\schemabox{#1}}{\schemabox{#2}}
}

```



```

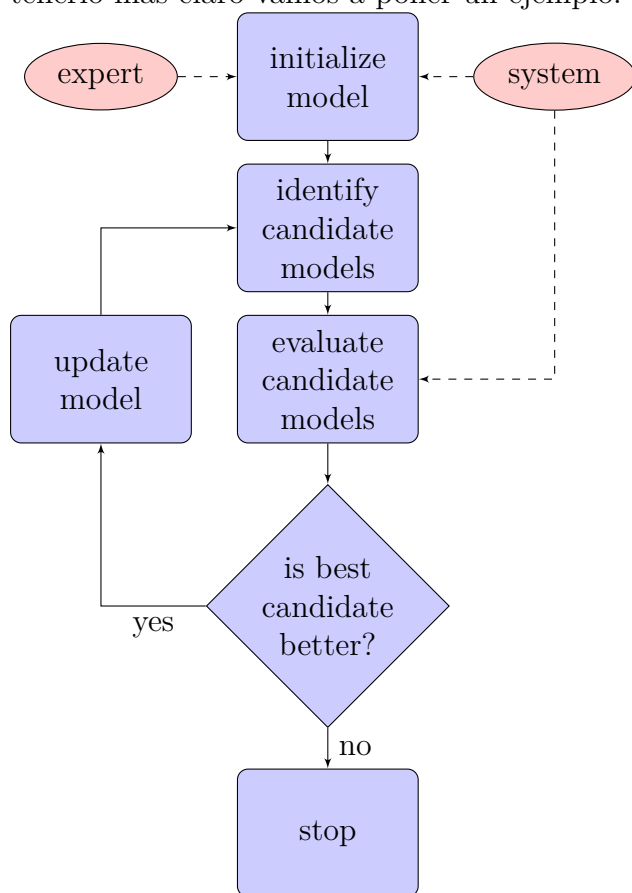
4 \diagram{\textbf{Criterios de exigencia}}
5 {- Constancia: \textbf{Trabajar mucho} \\
6 - Tiempo: \textbf{El tiempo empleado} \\
7 - \diagram{Otros}{- Buena presentación \\ - \ldots}}

```

### 7.3. Diagramas de flujo

Para realizar un diagrama de flujo tendremos que instalar el paquete `tikz` y la librería `\usetikzlibrary{shapes, arrows}` para que nos pueda crear los nodos con sus formas y flechas.

Una vez dentro del documento, utilizaremos el entorno `tikzstyle`, en el que determinaremos la forma de cada uno de los nodos y su distancia entre ellos. Dentro del documento, empezaremos con el entorno `tikzpicture` para dar características a los nodos, para ello utilizaremos el comando `\begin{tikzpicture}` para comenzar y para describir los nodos utilizaremos el comando `\node[tipo de bloque]{nombre}`, uniremos cada nodo con el comando `\path` y terminaremos nuestro entorno con el comando `\end{tikzpicture}`. Para tenerlo más claro vamos a poner un ejemplo:



Listing 7.5: Código diagrama de flujo

```

1 \documentclass { article }
2 \usepackage [ latin 1 ] { inputenc }
3 \usepackage { tikz }

```

```

4 \usetikzlibrary {shapes , arrows}
5 \usepackage {verbatim}
6 \usepackage [ active , tightpage ] {preview}
7 \PreviewEnvironment {tikzpicture}
8 \setlength \PreviewBorder {5pt} %
9 %% %
10 \begin {document}
11 \pagestyle {empty}
12
13
14 % Estilo bloques
15 \tikzstyle {decision} = [diamond, draw, fill=blue!20,
16     text width=4.5em, text badly centered, node distance=3cm,
17     inner sep=0pt]
18 \tikzstyle {block} = [rectangle, draw, fill=blue!20,
19     text width=5em, text centered, rounded corners, minimum
20     height=4em]
21 \tikzstyle {line} = [draw, -latex']
22 \tikzstyle {cloud} = [draw, ellipse, fill=red!20, node distance
23     =3cm,
24     minimum height=2em]
25
26 \begin {tikzpicture} [node distance = 2cm, auto]
27     % Nodos
28     \node [block] (init) {initialize model};
29     \node [cloud, left of=init] (expert) {expert};
30     \node [cloud, right of=init] (system) {system};
31     \node [block, below of=init] (identify) {identify
32         candidate models};
33     \node [block, below of=identify] (evaluate) {evaluate
34         candidate models};
35     \node [block, left of=evaluate, node distance=3cm] (update
36         ) {update model};
37     \node [decision, below of=evaluate] (decide) {is best
38         candidate better?};
39     \node [block, below of=decide, node distance=3cm] (stop) {
40         stop};
41
42     % Nexos unión
43     \path [line] (init) -- (identify);
44     \path [line] (identify) -- (evaluate);
45     \path [line] (evaluate) -- (decide);
46     \path [line] (decide) -| node [near start] {yes} (update);
47     \path [line] (update) |- (identify);
48     \path [line] (decide) -- node {no} (stop);
49     \path [line, dashed] (expert) -- (init);
50     \path [line, dashed] (system) -- (init);
51     \path [line, dashed] (system) |- (evaluate);

```

```
43 \end{tikzpicture}
44 \end{document}
```

## 7.4. Cambiar nombre de los capítulos con `renewcommand`

Cambiar el nombre de los capítulos o de las secciones se realiza de una manera sencilla, para ello solo tendremos que utilizar el comando `\renewcommand{\chapter}{Nuevo nombre}`. Y si queremos pasar los nombres de los capítulos a español utilizaremos el comando `\addto\captionsspanish{Nuevos nombres comandos}`. Se va a realizar de una manera muy sencilla y para ello, vamos a poner un ejemplo:

Listing 7.6: Código cambio de nombre

```
1 \documentclass {book}
2 \usepackage [utf8] {inputenc}
3 \usepackage [spanish] {babel}
4 \addto\captionsspanish {
5 \renewcommand {\partname} {Libro}
6 \renewcommand {\chaptername} {Jornada}
7 \renewcommand {\chapter} {Anexo}
8 }
9
10 \title {Libro}
11 \author {David Pacios}
12
13 \begin {document}
14
15 \maketitle
16 \chapter {Titulo}
17 \chaptername
18
19 \end {document}
```

## 7.5. Cambiar color de fondo de folio

Cambiar el color de fondo de folio se realiza de una manera muy sencilla, para ello sólo tendremos que utilizar el comando `\pagecolor{Nombre del color}` o mediante el comando `\pagecolor[modelo]{codigo}` por si lo queremos ponere de una manera más sencilla.

Figura 7.3: Fondo cambiado de color

Listing 7.7: Código cambio de color de fondo

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3 \usepackage [ spanish ] { babel }
4 \usepackage { color }
5 \pagecolor { blue }
6 \title { Articulo fondo }
7 \author { David Pacios }
8 \date { \today }
9 \begin { document }
10 \maketitle
11 \section { Fondo azul }
12 \end { document }
```

# Capítulo 8

## Capitulo 8

### 8.1. El entorno `verbatim`

El entorno que por defecto utiliza  $\LaTeX$  para mostrar el código es el `verbatim`. Genera una salida de código monoespaciada. Para comenzar este entorno comenzaremos con el comando `\begin{verbatim}` y terminaremos con el comando `\end{verbatim}`, y entre medias de ambos colocaremos nuestra línea de código. Para tenerlo más claro vamos a poner un ejemplo:

Este texto mantiene los comandos `\textbf{variados}` y se ignoran `\LaTeX{}`.

Listing 8.1: Código `verbatim`

```
1 \begin{verbatim}
2 Este texto mantiene los comandos \textbf{variados}
3 y se ignoran \LaTeX{ }.
4 \end{verbatim}
```

Además, si queremos colocar una línea de código únicamente o destacar cualquier comando, utilizaremos el comando `\verb|` y entre barras como esta `|` colocaremos nuestro comando. Para tenerlo más claro vamos a poner un ejemplo:  
Este comando `D: Trabajo\Carpeta` indica la ubicación.

Listing 8.2: Código `verbatim` texto

```
1 Este comando \verb|D: Trabajo\Carpeta| indica la ubicación.
```

### 8.2. Usando `listings` para resaltar el código

Otro entorno que tenemos para presentar código es el `listings`, es muy importante definir en el preámbulo el paquete `listings` y una vez hecho, se comenzará con el comando `\begin{lstlisting}[lenguaje, número de línea]`, entre corchetes colocaremos el lenguaje que estamos utilizando, y si nos dividen el código indicaremos la línea, y terminaremos con el comando `\end{lstlisting}`. Para tenerlo más claro vamos a poner un ejemplo:

```
1 \begin{frame}{Cryptoparty}
2   \begin{alertblock}{Evento Internacional}
3     Organizado por la UCM en España, este 6 de Abril en el Cí
4       rculo de Bellas artes. \url{https://cryptoparty.ucm.es/}
5   \end{alertblock}
6   \begin{exampleblock}{¿Qué es?}
7     Las CryptoParties son un evento gratuito y abierto para
8       todo el mundo, especialmente para aquellos sin
9       conocimientos previos que no hayan asistido previamente.
10  \end{exampleblock}
11  \begin{center}
12    \includegraphics [width=0.2\textwidth] {Figures/Cryptoparty
13      _2019.jpg}
```

```
16   \end{center}
17 \end{frame}
```

Entre los lenguajes que podemos utilizar se encuentran los siguientes:

- C++.
- Cobol.
- Gnupot.
- HTML.
- Octave.
- Pascal.
- Python.
- Scilab.
- VHDL.
- XML.

### 8.3. Importando el código

Además de poder pegar nuestro código en el [listing](#) directamente, también podremos importar nuestro código con el comando `\lstinputlisting[lenguaje]{Programa}`. Para verlo más claro vamos a poner un ejemplo:

```

1 X=[
    365.015,404.64,407.55,435.91,546.01,576.93,579.13,696.50,706
    ]; %valores de lamba experimental
2 Y=[365.015, 404.656, 407.783, 435.833, 546.074, 576.960,
    579.066,696.543,]; %valores de lamba teórica
3 N=19; %número de puntos del ajuste
4
5
6 %Calculo de la penmdiente y la ordenada en el origen con su
    error
7 xm=mean(X);ym=mean(Y);
8 sumxx=sum((X-xm).^2);sumxy=sum((X-xm).*(Y-ym));sumyy=sum((Y-ym
    ).^2);
9 a=sumxy/sumxx
10 b=ym-a*xm
11 d=Y-a*X-b;

```

Listing 8.3: Código de importar el código

```

1 \lstinputlisting [language=Octave, firstline=2, lastline=12] {
    regresionTFG.m}

```

## 8.4. Dando estilo al código

Este entorno se puede modificar en el preámbulo los siguientes parámetros:

- **backgroundcolor**: Indica el color de fondo. Necesita el paquete [color](#) o [xcolor](#).
- **commenstyle**: Estilo de los comentarios en el lenguaje.
- **basicstyle**: Fuente, tamaño de la letra en el código.
- **keywordstyle**: Estilo de las palabras clave.
- **numberstyle**: Estilo de las numeraciones.
- **numbersep**: Distancia entre los números del código.
- **stringstyle**: Estilo de las cadenas en el lenguaje.
- **showspaces**(true/false): Enfatiza en los espacios de las cadenas.
- **showstringspaces**(true/false): Enfatiza en los espacios en las cadenas.
- **showtabs**(true/false): Enfatiza en las tabulaciones en el código.
- **numbers**(left/right/none): Posición de los números.
- **prebreak**: Indica una marca al terminar una línea.

- **captionpos** (t/b): Posición del caption.
- **frame** (none/leftline/topline/bottomline/lines/single/shadowbox): Muestra el marco fuera del código.
- **breakwhitespace**: Muestra los espacios que ocurren cuando hay espacios en blanco.
- **breaklines**: Saltos de línea automáticos.
- **keepspaces**: Mantiene los espacios en el código, es útil para la indentación.
- **tabsize**: Tamaño de tabla por defecto.
- **escapeinside**: Especifica algunos caracteres en el código.
- **rulecolor**: Especifica el color en el marco de la caja.

Listing 8.4: Código ejemplo personalizado

```
1 \documentclass { article }
2 \usepackage [ utf8 ] { inputenc }
3
4 \usepackage { listings }
5 \usepackage { color }
6 %Colores definidos
7 \definecolor { codegreen } { rgb } { 0 , 0.6 , 0 }
8 \definecolor { codegray } { rgb } { 0.5 , 0.5 , 0.5 }
9 \definecolor { codepurple } { rgb } { 0.58 , 0 , 0.82 }
10 \definecolor { backcolour } { rgb } { 0.95 , 0.95 , 0.92 }
11
12 %Dandole estilo al codigo
13 \lstdefinestyle { mystyle } {
14   backgroundcolor = \color { backcolour } ,   commentstyle = \color {
15     codegreen } ,
16   keywordstyle = \color { magenta } ,
17   numberstyle = \tiny \color { codegray } ,
18   stringstyle = \color { codepurple } ,
19   basicstyle = \footnotesize ,
20   breakatwhitespace = false ,
21   breaklines = true ,
22   captionpos = b ,
23   keepspaces = true ,
24   numbers = left ,
25   numbersep = 5pt ,
26   showspaces = false ,
27   showstringspaces = false ,
28   showtabs = false ,
29   tabsize = 2
```



```

30 }
31 %Set de mi código
32 \lstset{style=mystyle}

```

Además de lo explicado anteriormente, podemos darle estilo con los siguientes comandos:

- `\lstdefinestyle{mystyle}{}`: Definimos el código llamándolo `mystyle` y le damos las características que queramos.
- `\lstset{stylemystyle}`: Estilo ya definido.

## 8.5. Listado y nombre del código

Como en las imágenes y en las tablas, podremos darle nombre y numerar nuestro código con el comando `caption` al lado del lenguaje. Para tenerlo más claro vamos a poner un ejemplo:

Listing 8.5: Ejemplo con nombre

```

1 import numpy as np
2
3 def incmatrix(genl1, genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m, 1), int) #dummy variable

```

```

28 #compute the bitwise xor matrix
29 M1 = bitxormatrix(genl1)
30 M2 = np.triu(bitxormatrix(genl2), 1)
31
32 for i in range(m-1):
33     for j in range(i+1, m):
34         [r, c] = np.where(M2 == M1[i, j])
35         for k in range(len(r)):
36             VT[(i)*n + r[k]] = 1;
37             VT[(i)*n + c[k]] = 1;
38             VT[(j)*n + r[k]] = 1;
39             VT[(j)*n + c[k]] = 1;
40
41         if M is None:
42             M = np.copy(VT)
43         else:
44             M = np.concatenate((M, VT), 1)
45
46     VT = np.zeros((n*m, 1), int)
47

```

```
48      return M
```

Listing 8.6: Ejemplo con nombre código

```

1  \begin{lstlisting}[language=Python, caption=Ejemplo con nombre
   ]
2  import numpy as np
3
4  def incmatrix(genl1, genl2):
5      m = len(genl1)
6      n = len(genl2)
7      M = None #to become the incidence matrix
8      VT = np.zeros((n*m,1), int) #dummy variable
9
10     #compute the bitwise xor matrix
11     M1 = bitxormatrix(genl1)
12     M2 = np.triu(bitxormatrix(genl2), 1)
13
14     for i in range(m-1):
15         for j in range(i+1, m):
16             [r, c] = np.where(M2 == M1[i, j])
17             for k in range(len(r)):
18                 VT[(i)*n + r[k]] = 1;
19                 VT[(i)*n + c[k]] = 1;
20             VT[(j)*n + r[k]] = 1;
21             VT[(j)*n + c[k]] = 1;
22
23             if M is None:
24                 M = np.copy(VT)
25             else:
26                 M = np.concatenate((M, VT), 1)
27
28             VT = np.zeros((n*m,1), int)
29
30     return M

```

Y si queremos que nos aparezca en una lista utilizaremos el comando `\lstlistoflistings`.

## 8.6. Poner palabras clave

Para poner palabras clave utilizaremos el comando `\lstdefinlanguage{lenguaje}{Características}`. Para tenerlo más claro vamos a poner un código de ejemplo:

Listing 8.7: Código palabras clave

```

1  \lstset{language=LaTeX,
2      keywordstyle=\color{rojo},
3      texcsstyle=*\color{myblue},

```

```

4   basicstyle=\textbf\normalfont\ttfamily ,
5   commentstyle=\color{comments}\ttfamily ,
6   stringstyle=\rmfamily ,
7   numbers=left ,
8   numberstyle=\scriptsize ,
9   stepnumber=1,
10  numbersep=8pt ,
11  captionpos=top ,
12  showstringspaces=false ,
13  breaklines=true ,
14  frameround=ftff ,
15  morekeywords={RequirePackage , ProvidesPackage ,
16               NeedsTeXFormat} ,
17  backgroundcolor=\color{background} ,
18  literate=
19      *{\{\}\{\textcolor{myblue}\{\}\}\{1}
20       {\}\{\textcolor{myblue}\}\}\{1}
21       {\}\{\textcolor{myblue}\textbackslash}\}\{1}
22       {\$}\{\textcolor{rojo}\$\}\{1}
23       {\&}\{\textcolor{rojo}\&}\}\{1}
24       {\documentclass}\{\textcolor{rojo}\textbackslash
25        \documentclass}\}\{12}
26       {\%\\ \documentclass}\{\textcolor{rojo}\
27        textbackslash \documentclass}\}\{12}
28 {\%\\usepackage}\{\textcolor{rojo}\textbackslash usepackage}
29 }\{9}
30
31     {\}\{\textcolor{myblue}\{\}\}\{1}
32     {\}\{\textcolor{myblue}\}\}\{1}
33     {á}\{'a}\}1 {é}\{'e}\}1 {í}\{'i}\}1 {ó}\{'o}\}1 {ú
34     }\{'u}\}1 {ñ}\{\~n}\}1 {Á}\{'A}\}1 {É}\{'E}\}1 {Í
35     }\{'I}\}1 {Ó}\{'O}\}1 {Ú}\{'U}\}1,
36
37  frame=single ,
38  frameround={t}{t}{t}{t},
39  framexleftmargin=6mm,
40  numbers=left ,
41  numberstyle=\tiny\color{halfgray} ,
42 }

```



# Capítulo 9

## Capitulo 9

### 9.1. Bibliografía

La bibliografía[?] es una lista que aparece al final[?] de nuestro trabajo y recopila, de una forma ordenada, alfabéticamente, la información completa de todas las fuentes citadas, como de las que han sido buscadas durante su preparación.[?] Y la estructura de la bibliografía es la siguiente:[?]

- Empezamos primero con el apellido del o de los autores, presentando primero el apellido y luego las iniciales.
- Si no tiene autor, citamos el título y se ordena alfabéticamente con la primera palabra significativa del título.
- Si tenemos más de una referencia del mismo autor, lo ordenaremos de forma cronológica, comenzamos con la publicación más antigua.
- La lista que tendremos tendrá un solo orden.
- Cada nueva referencia la colocamos en una línea nueva.
- Las referencias no pueden ser numeradas.

Para realizar una bibliografía de una manera sencilla, vamos a utilizar el entorno [thebibliography](#). Este entorno es preferible usarlo cuando lleva pocas citas bibliográficas o cuando no tengamos una biblioteca donde podamos guardar todas las referencias bibliográficas. La bibliografía se colocará directamente en el documento, para ello, nosotros mismos colocaremos por orden todos los elementos necesarios. El enumerado y el encabezado “Referencias” son creados de manera automática.

# Bibliografía

David Pacios

September 2018

## Referencias

- [1] GUTERMAN TULLIO *Informática y deporte* Colección Gestión Deporte
- [2] ALONSO RIVAS, GONZALO *Auditoria informática*, 1989

Figura 9.1: Ejemplo de bibliografía

Listing 9.1: Ejemplo de bibliografía

```
1 \title{Bibliografía}
2 \author{David Pacios}
3 \date{September 2018}
4 \begin{document}
5 \maketitle
6 \begin{thebibliography}{}
7 \bibitem{Guter}\textsc{Guterman Tulio}
8 \textit{Informática y deporte} Colección Gestión Deporte
9 \bibitem{Alon}\textsc{Alonso Rivas , Gonzalo}
10 \textit{Auditoria informática} , 1989
11 \end{thebibliography}
12 \end{document}
```

Es muy importante saber cómo referenciar la bibliografía, ya que, si no referenciamos bien, nos pueden acusar de plagio.

Por otro lado, podemos utilizar el comando `\cite{}` para citar la cita en alguna parte del texto. Para verlo más claro, vamos a realizar una cita del ejemplo anterior.

# Bibliografía

David Pacios

September 2018

## Referencias

[1] GUTERMAN TULIO *Informática y deporte* Colección Gestión Deporte

[2] ALONSO RIVAS, GONZALO *Auditoria informática*, 1989

### 1. Citar bibliografía

Este libro[1] es utilizado en el INEM.

Figura 9.2: Referencia citada

Listing 9.2: Ejemplo de cita

```
1 \title{Bibliografía}
2 \author{David Pacios}
3 \date{September 2018}
4 \begin{document}
5 \maketitle
6 \begin{thebibliography}{}
7 \bibitem{Guter}\textsc{Guterman Tulio}
8 \textit{Informática y deporte} Colección Gestión Deporte
9 \bibitem{Alon}\textsc{Alonso Rivas, Gonzalo}
10 \textit{Auditoria informática}, 1989
11 \end{thebibliography}
12 \section{Citar bibliografía}
13 Este libro\cite{Guter} es utilizado en el INEM.
14 \end{document}
```

#### 9.1.1. Estilos de bibliografías

Los estilos son un conjunto de reglas que definen qué datos deben incluirse al citar cada referencia, tanto el orden y el formato tipográfico.

La regulación de referencias bibliográficas está regulada a nivel internacional por la norma **ISO 690:2010**; y en España, está regulada por la norma **UNE-ISO 690:2013**.

Para saber qué estilo bibliográfico utilizar vamos a tener en cuenta:

- Si estamos ante un trabajo académico, pediremos consejo a nuestro tutor.

- Si vamos a realizar una publicación de un libro o una revista, la editorial nos guiará para realizar las referencias.

De todos los estilos bibliográficos, vamos a destacar los siguientes:

- **APA:** Creado en 1929, es el estilo American Psychological Association. Es utilizada principalmente en psicología y pedagogía.
- **CHICAGO:** Publicado por primera vez en 1903. Es utilizada en las áreas de ciencias sociales y humanas.
- **MLA:** Es empleada principalmente en humanidades, letras y lingüística.
- **HARVARD:** Es un estilo que sigue los formatos, colocando primero el autor-fecha, y de forma ordenada, al final se encuentra el año de la publicación. Presenta variaciones en su aplicación.

### 9.1.2. BibTeX

BibTeX es una herramienta hermana de L<sup>A</sup>T<sub>E</sub>X especializada en el apoyo de la bibliografía. Está recomendada para bibliografías muy extensas o para citas que sean nombradas muy frecuentemente.

Para ello, necesitamos una biblioteca virtual, que es un archivo con extensión `.bib`, que contiene la bibliografía que requiere nuestro documento. Para utilizarla en la bibliografía vamos a utilizar el comando `\bibliography{}` y entre corchetes nombraremos nuestra biblioteca virtual.

Y para agregar los archivos a la bibliografía utilizaremos el comando `\cite{}`, y entre corchetes pondremos el nombre de nuestra referencia bibliográfica, si alguna entrada no se quiere citar utilizaremos el comando `\nocite{}`, y finalmente, por otro lado, si los queremos nombrar indiscriminadamente utilizaremos el comando `\nocite{*}`. Para tenerlo más claro, vamos a poner un ejemplo:



# Bibtex

David Pacios

September 2018

De esta manera podemos nombrar nuestra bibliografía[1]. De esta manera podemos añadir otra cita[2] y por aquí, la otra[3].

## Referencias

- [1] JosÚ Antonio Echenique García. Auditoría en informática. 2001.
- [2] Ian Hacking. *Representing and intervening*, volume 279. Cambridge University Press Cambridge, 1983.
- [3] Raúl González Gómez et al. Estudio y análisis de vulnerabilidades de la deep web mediante la implementación de un nodo tor. 2018.

Figura 9.3: Ejemplo BibTeX

Listing 9.3: Ejemplo de BibTeX

```
1 \title{Bibtex}
2 \author{David Pacios}
3 \date{September 2018}
4
5 \begin{document}
6 \maketitle
7 De esta manera podemos nombrar nuestra bibliografía\cite{
   echenique2001auditoria}. De esta manera podemos añadir otra
   cita\cite{hacking1983representing} y por aquí, la otra\cite{
   gonzalez2018estudio}.
8 \bibliographystyle{unsrt}
9 \bibliography{bibliografia}
```

Como podemos ver en el ejemplo anterior, podemos cambiar el estilo de nuestra bibliografía y es muy importante, crear en un documento aparte nuestra bibliografía, esta parte la explicaremos con detenimiento más adelante.

Además de todo lo anterior, podemos utilizar el paquete `cite` para contraer las referencias y se pueden utilizar más estilos de bibliografías como:

- `acm`: Ordenados alfabeticamente y referencias con números.
- `IEEEannot`: Es similar al anterior, pero no pone los autores en mayúsculas.

- `unsrt`: Ordena los artículos por orden de aparición y los referencia con números.
- `jtbnew`: Se utiliza para hacer referencias con nombre y año. En este estilo es mejor nombrarlas por separado.

Para tenerlo más claro, vamos a poner otro ejemplo cambiando el estilo de la bibliografía:

## Bibtex

David Pacios

September 2018

De esta manera podemos nombrar nuestra bibliografía[1]. De esta manera podemos añadir otra cita[3] y por aquí, la otra[2].

### Referencias

- [1] ECHENIQUE GARCÍA, J. A. Auditoría en informática.
- [2] GONZÁLEZ GÓMEZ, R., ET AL. Estudio y análisis de vulnerabilidades de la deep web mediante la implementación de un nodo tor.
- [3] HACKING, I. *Representing and intervening*, vol. 279. Cambridge University Press Cambridge, 1983.

Figura 9.4: Ejemplo BibTex 2

Listing 9.4: Ejemplo de BibTex 2

```
1 \title{Bibtex}
2 \author{David Pacios}
3 \date{September 2018}
4 \begin{document}
5 \maketitle
6 De esta manera podemos nombrar nuestra bibliografía\cite{
   echenique2001auditoria}. De esta manera podemos añadir otra
   cita\cite{hacking1983representing} y por aquí, la otra\cite{
   gonzalez2018estudio}.
7 \bibliographystyle{acm}
8 \bibliography{bibliografia}
```

Una vez que tengamos claro lo anterior, en el siguiente capítulo vamos a ver cómo crear nuestra bibliografía. Es muy importante crear previamente nuestra biblioteca `.bib`.

## Crea tu propia referencia bibliográfica

Para empezar vamos a crear una bibliografía donde poner nuestras referencias, como por ejemplo, `bibliografia.bib` donde vamos a colocar nuestros artículos de referencia. En nuestro documento se vería de la siguiente manera:

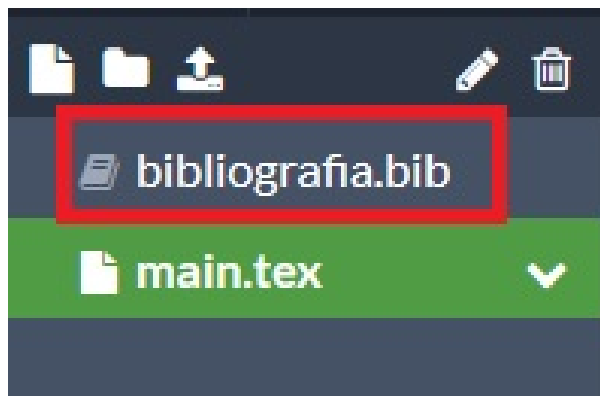


Figura 9.5: Colocación bibliografía

Una vez tenemos claro donde colocar nuestra bibliografía, vamos a ver cómo tenemos que poner nuestra propia bibliografía. Aquí tenemos dos opciones:

- Utilizar Google Scholar para que lo traduzca directamente.
- Utilizar directamente nosotros mismos los distintos comandos.

Dentro de estos comandos dispondremos de campos requeridos (obligatorios) y opcionales. Una vez lo tenemos claro, vamos a ver los tipos de entradas con sus distintos campos:

- **ARTICLE**: Un artículo.
  - **Campos requeridos**: author, title, journal, year.
  - **Campos opcionales**: volume, number, pages, month, note.
- **BOOK**: Un libro con editorial específica.
  - **Campos requeridos**: author o editor, title, publisher, year.
  - **Campos opcionales**: volume o number, series, address, edition, month, note.
- **PHDTHESIS**: Tesis doctorado.
  - **Campos requeridos**: author, title, school, year.
  - **Campos opcionales**: type, address, month, note.

Finalmente, una vez hemos visto los tipos de entrada, vamos a ver qué significa los campos descritos anteriormente:

- **address**: Dirección de la editorial.
  - **author**: Nombre(s) de (de los ) autor (es).
  - **booktitle**: Título del libro.
-

- [edition](#): Edición del libro.
- [pages](#): Número de páginas.
- [year](#): Año de la publicación.

Para tenerlo más claro, vamos a ver el código de la bibliografía del ejemplo anterior:

Listing 9.5: Código bibliografía ejemplo 2

```
1 @article{echenique2001auditoria ,
2   title={Auditor{\ 'i}a en inform{\ 'a}tica.},
3   author={Echenique Garc{\ 'i}a, Jos{\ 'U} Antonio},
4   year={2001},
5   publisher={McGraw- Hill}
6 }
7 @book{hacking1983representing ,
8   title={Representing and intervening},
9   author={Hacking, Ian},
10  volume={279},
11  year={1983},
12   publisher={Cambridge University Press Cambridge}
13 }
14 @article{gonzalez2018estudio ,
15   title={Estudio y an{\ 'a}lisis de vulnerabilidades de la Deep
16     Web mediante la implementaci{\ 'o}n de un nodo Tor},
17   author={Gonz{\ 'a}lez G{\ 'o}mez, Ra{\ 'u}l and others},
18   year={2018}
19 }
```

## 9.2. Uso de Google Scholar

Es un buscador especializado que permite localizar documentos académicos como artículos, tesis, libros, patentes, materiales de congresos y resúmenes de universidad u otras organizaciones académicas.

Nos permite acceder al documento del texto entero o al abstract del texto.

Figura 9.6: Buscador Google Scholar

Los resultados de la búsqueda se muestran de la siguiente manera:

- [\[CITAS\]](#): Referencias bibliográficas.
- [\[LIBRO\]](#): Referencia de un libro.
- [\[PDF\]](#): Documento en formato pdf.
- [\[DOC\]](#): Documento en formato word.

- [\[HTML\]](#): Documento en formato web.
- El número de veces que ha sido citado.
- [cited by](#): Indica el número de citas que ha recibido y por quién.
- [related articles](#): Muestra una relación de documentos según temática y relevancia.

Figura 9.7: Búsqueda en Google Scholar

Una vez sabemos cómo exponen los resultados de la búsqueda, vamos a ver cómo realizar una cita mediante Google Scholar:

- Seleccionamos el artículo.
- Buscamos en la parte inferior la palabra [Citar](#) o [Importar al BibTeX](#).
- Seleccionamos cualquiera de los dos y nos aparecerá la cita.

[DOC] Auditoria informática

[DOC] itla.edu.do

M Piattini, E Del Peso - Un enfoque práctico, 2001 - itla.edu.do

En la actualidad los temas relativos a la auditoría **informática** cobran cada vez más relevancia, tanto a nivel nacional como internacional, debido a que la información se ha convertido en el activo más importante de las empresas, representando su principal ventaja ...

☆ 99 **Citado por 127** Artículos relacionados Las 5 versiones **Importar al BibTeX** Versión en HTML

Figura 9.8: Citar Google Scholar

Una vez hemos seleccionado cualquiera de los dos nos aparecerá lo siguiente:

```
@article{piattini2001auditoria,
  title={Auditoria inform{'a}tica},
  author={Piattini, Mario and Del Peso, Emilio},
  journal={Un enfoque pr{'a}ctico},
  volume={6},
  year={2001}
}
```

Figura 9.9: Citación en BibTeX de Google Scholar

Con todo esto, podremos insertar en nuestra bibliografía nuestras citas.



# Capítulo 10

## Plantilla básica para escritores

### 10.1. Introducción

La redacción de un buen documento es muy importante tanto para escritores como para estudiantes. En los siguientes capítulos se señalarán los errores gramaticales y de escritura comunes y se verán como corregirlos.

### 10.2. La puntuación

La puntuación es esencial para realizar una buena narración, para ello vamos a repasar las reglas para cada signo de puntuación.

#### 10.2.1. La coma

##### Regla esencial

Nunca se debe separar con una coma el sujeto del verbo.

##### Ejemplos de frases mal construidas:

Yo, voy en transporte público a estudiar.  
Mi padre, se compró un coche rojo.

Como excepción, si el verbo se repite, entonces si separaremos con una coma el sujeto del predicado:

##### Ejemplos de frases bien construidas:

Tres por cinco, quince.  
Tres por cinco son quince.

##### Primera Regla

No se pone coma antes de la conjunción y en una enumeración.

La coma separa los elementos en una enumeración excepto en su último elemento:

**Ejemplos de enumeraciones:**

Compró peras, manzanas, y fresas.  
Compró peras, manzanas y fresas.

Como toda regla, tiene excepciones, en este caso, tenemos las siguientes:

**Primera excepción:** Poner énfasis en el último de la enumeración, siempre que no tenga relación con las enumeraciones anteriores.

**Ejemplos de frases bien construidas:**

Le regalaron unos libros, una bicicleta, y una consola.  
Compramos unas frutas, verduras, y se le cayeron las frutas.

**Segunda excepción:** Si la siguiente idea significativa que se escribe después de una enumeración no califica el último elemento y es independiente de la idea anterior.

**Ejemplo de frase bien construida:**

Comimos una pizza, patatas, sopa y fruta, y nos fuimos a pescar.

**Segunda Regla**

Los vocativos y las aposiciones siempre necesitan una coma.

Un vocativo sirve para nombrar o llamar la atención de una persona.

**Ejemplos de frases bien construidas:**

¿Qué pasa, Juan?  
José Luis, cállate un rato.  
Espera, Laura, vuelve a leer lo anterior.

Las aposiciones son explicaciones que se intercalan en la frase. Se pueden explicar también por medio de rayas o de paréntesis.

**Ejemplos de frases bien construidas:**

Luisa, la que se sienta primero en clase, ha sacado un diez.  
Londres, capital de Reino Unido, no quiere salir de la UE.

### 10.3. El punto y la coma

Este signo de puntuación sirve para insertar una pausa mayor que la de la coma, pero menor que la de un punto.

Se puede utilizar para enumeraciones más largas o complejas, como por ejemplo:

**Ejemplo de frase bien construida:**

Pon los paquetes sobre la mesa; la fruta y la verdura dentro del bol; los vasos en la



encimera, y los bollos, en la bandeja.

También podemos utilizarlo para separar frases independientes que están relacionadas entre sí. Dependiendo del grado de vinculación entre las frases valoraremos si utilizar, por un lado, si el vínculo es débil deberíamos usar un punto, y por otro lado, si el vínculo es fuerte deberíamos usar el punto y coma.

#### Ejemplos de frases bien construidas:

Han suspendido muchos; no podrán irse de viaje de fin de curso.  
Han suspendido muchos. No podrán irse de viaje de fin de curso.  
Ha suspendido muchos: no podrán irse de viaje de fin de curso.

### 10.3.1. El punto

El punto es el signo de puntuación que indica una pausa mayor. Su uso principal es el de señalar el final de una frase, de un párrafo o de un texto.

Si el punto se encuentra al final de un enunciado tras el que continúa el mismo párrafo, se llama punto y seguido; si se encuentra al final de un párrafo, se llama punto y aparte; y si se encuentra al final del texto, se llama punto final.

#### Primera Regla

Siempre se escribe detrás de las comillas, de los paréntesis y de las rayas de cierre.

#### Ejemplos de frases bien construidas:

El abuelo puede que estuviera de mal humor (puede que le hubieran dicho algo mal).  
- Hola a todos- dijeron los actores al empezar la función-. ¿Qué tal están?

#### Segunda Regla

Nunca se pone un punto después de los signos de cierre de interrogación ni exclamación.

#### Ejemplos de frases mal construidas:

¿Queda mucho?.  
¡Estoy harto!. ¡No toques más el claxon!

#### Ejemplos de frases bien construidas:

¿Queda mucho?  
¡Estoy harto! ¡No toques más el claxon!

#### Tercera Regla

En los años escritos con cifras no se escribe punto.

**Ejemplos de frases bien construidas:**

El año 2012 estuvo lleno de tragedias.  
Este libro tiene un total de 1500 páginas.

**Cuarta Regla**

Los números escritos con cifras que excedan el millar tampoco se separarán con un punto.

Se recomienda separar los millares con espacios en grupos de tres, empezando por la derecha:

**Ejemplo de frase bien construida:**

La población mundial está cerca de los 10 000 000 millones de habitantes.

### 10.3.2. Los dos puntos

Los dos puntos representan una pausa mayor que la de la coma y menor que la del punto. Esta pausa ofrece una explicación de lo que acaba de decirse o una enumeración:

**Ejemplo de frase bien construida:**

He comprado de regalo: unos libros, una película y una revista.

Además, podemos escribir los dos puntos en el saludo inicial de una carta.

**Ejemplo de frase bien construida:**

Queridos alumnos:  
Les escribo para informaros...

Por otro lado, tenemos que es incorrecto escribirlo entre una preposición y el sustantivo o sustantivos que están a continuación:

**Ejemplo de frase mal construida:**

Las capitales mundiales son: Londres, Madrid, París y Bélgica.

**Ejemplo de frase bien construida:**

Las capitales mundiales son Londres, Madrid, París y Bélgica.

Y finalmente, sólo escribiremos en mayúsculas después de una cita o palabras textuales.

**Ejemplo de frase bien construida:**

La profesora dijo: “Hay que hacer todos los deberes”.

### 10.3.3. Las comillas

Las comillas se utilizan para reproducir el pensamiento de un personaje en novelas, libros o cuentos. Su contenido no debe ser escrito en cursiva.

#### Ejemplos de frases mal construidas:

*Tengo que ir al gimnasio, pensó Luis.*  
*Tengo que ir al gimnasio, pensó Luis.*  
*“Tengo que ir al gimnasio”, pensó Luis.*

#### Ejemplo de frase bien construida:

“Tengo que ir al gimnasio”, pensó Luis.

Si insertamos un comentario del narrador en el pensamiento, debemos enmarcarlo entre rayas. No hace falta cerrar las comillas y volver a abrirlas:

#### Ejemplo de frase mal construida:

“Debería llegar a tiempo”- enunció en alto- “a ver si llega el metro a su hora”.

#### Ejemplo de frase bien construida:

“Debería llegar a tiempo- enunció en alto- a ver si llega el metro a su hora”.

Si un personaje vuelve a decir lo mismo, y lo que dice ocupa varios párrafos, debemos colocar unas comillas de cierre al principio de cada uno de ellos y cerrar el último con las comillas de cierre.

Además, podremos utilizar las comillas para frases o expresiones con ironía o con un sentido diferente al original.

#### Primera Regla

Las comillas angulares son las propias del español.

Cuadro 10.1: Tipos de comillas

Tipos de comillas	Comillas
Angulares o españolas	« »
Inglesas	“ ”
Simples	‘ ’

Para escribir las comillas angulares o españolas en Windows utilizaremos el comando [Alt+0171](#) para las comillas de apertura, y [Alt+0187](#).

#### Segunda Regla

Se recomienda usar las comillas angulares primero, luego las inglesas y finalmente las simples.

**IMPORTANTE:** Las comillas se escriben pegadas a la primera y la última palabra del segmento que entrecomillan.

**Tercera Regla**

Cuando el texto entrecomillado cierra la oración, siempre se pone el punto después, no antes.

Si un texto entrecomillado acaba con un signo de interrogación y exclamación, después tenemos que poner el punto.

**Ejemplos de frases bien construidas:**

La profesora explicó: « Uno más uno son dos».  
«¡Qué horror!», pensó mientras se asustaba.

## 10.4. Los signos de interrogación y exclamación

**Regla esencial**

Siempre hay que escribir el signo de apertura.

Es muy importante poner signos de apertura, aunque estemos acostumbrados a ignorarlos en nuestra vida diaria.

**Ejemplos de frases mal construidas:**

Qué tal estás?  
Qué susto!

**Primera Regla**

Los signos de puntuación deben ir donde empieza la pregunta o la exclamación.

**Ejemplos de frases mal construidas:**

¿Si te dieran una beca, te irías a estudiar fuera?  
¡Qué inspiración, el resultado fue óptimo!

**Ejemplos de frases bien construidas:**

Si te dieran una beca, ¿te irías a estudiar fuera?  
¡Qué inspiración!, el resultado fue óptimo.

**Tercera Regla**

Si una interrogación o una exclamación van precedidas por un pero, no se pone coma antes de los signos de apertura.

**Ejemplo de frase bien construida:**

Pero ¡que mal está montado!

## 10.5. El guión

El guión se utiliza para unir palabras que están relacionadas entre sí o forman palabras compuestas.

### Ejemplos de palabras bien construidas:

Kilómetros-hora.  
Dirección sur-este.  
Obra hispano-alemana.

### 10.5.1. Los puntos suspensivos

#### Regla esencial

Los puntos suspensivos solo son tres.

Se usan para indicar vacilación o pausar e interrupciones en oraciones o discursos.

### Ejemplos de frases bien construidas:

Me gustaría conocer... No tengo mucha idea... ¿Sabes algún sitio chulo de vacaciones?  
Vete por ahí...

Además, se le pueden dar la intención de escribir alguna grosería o algún taco. Otro uso que se le puede dar es cambiando la palabra etcétera por los tres puntos. Es muy importante no colocar los puntos suspensivos con etcétera. Otra regla importante a tener en cuenta, es que, tras los signos de puntuación no se deja espacio de separación.

### 10.5.2. La raya y los paréntesis

Las rayas se utilizan para hacer una aclaración o un inciso en medio de una frase. Dependiendo del grado de alejamiento que queramos darle, usaremos de menor a mayor grado, las comas, las rayas o los paréntesis.

### Ejemplos de frases bien construidas:

Cristóbal Colón, de origen desconocido, descubrió América.  
Cristóbal Colón -de origen desconocido- descubrió América.  
Cristóbal Colón (de origen desconocido) descubrió América.

El paréntesis se usa normalmente para introducir fechas o lugares.

**IMPORTANTE:** Es incorrecto situar la aclaración después de una coma cuando queremos explicar en qué país se encuentra una ciudad.

**Ejemplo de frase mal construida:**

La ciudad más romántica se encuentra en París, Francia.

**Ejemplo de frase bien construida:**

La ciudad más romántica se encuentra en París (Francia).

**IMPORTANTE:** Es incorrecto usar la raya para separar el número de capítulo y su título.

**Ejemplo de palabras mal construidas:**

1-Introducción  
2-Definiciones

## 10.6. La raya en los diálogos

**Regla esencial**

La única raya válida para los diálogos en español es “—”.

Marca las intervenciones de cada personaje. Siempre se escribe pegada a la primera letra o al signo de puntuación.

**Ejemplos de diálogos mal contruidos:**

-¿Qué hora es?  
— ¿Qué hora es?

**Ejemplo de diálogo bien construido:**

—¿Qué hora es?

La raya del diálogo se escribe en Windows pulsando [Alt+0151](#).

Si el narrador realiza otro comentario, dejaremos un espacio antes de la raya.

**Ejemplo de diálogo bien construido:**

— Mañana realizaré el trabajo —dijo la profesora.

**Primera Regla**

Minúscula para los verbos de habla, mayúscula para el resto.

Si el narrador enuncia un verbo de habla, se enunciará en minúscula, pero si no es un verbo de habla, se empezará en mayúscula.

**Ejemplos de diálogos bien contruidos:**

— Puedo hacerlo. — Sorprendido, lo enunció—. Lo voy a aprobar—gritó.

## 10.7. Palabras esdrújulas y sobresdrújulas

**Regla esencial**

Siempre llevan tilde.

**Ejemplo de palabras esdrújulas y sobresdrújulas:**

Pétalo  
Mecánico  
Llévatelo

## 10.8. Otras normas

Los monosílabos nunca se acentúan, excepto cuando se trata de palabras que se escriben igual, pero que pertenecen a diferentes categorías gramaticales.

**Ejemplos de monosílabo con tilde:**

Posesivo/ mí.

**Ejemplo de monosílabo sin tilde:**

Pronombre/ mi.

Toda combinación de vocal abierta (a, e, o)+ vocal cerrada (i, u), o viceversa, siempre que la cerrada no sea tónica, así como la combinación de dos vocales cerradas distintas, han de considerarse diptongos.

Por otro lado, un gran grupo de formas verbales y palabras perdieron su tilde, pues pasaron a considerarse monosílabas.

**Ejemplos de monosílabos sin tilde:**

Fie  
Guio

La acentuación de los pronombres demostrativos éste, ése y aquél (tanto en masculino como en femenino, en singular como en plural) y del adverbio, sólo, únicamente cuando en una oración exista riesgo de ambigüedad porque puedan interpretarse de una u otra forma.

**Solo con tilde y sin tilde:**

Estaré estudiando solo una semana.

Estaré sólo un mes.

## 10.9. Los tiempos verbales en la narración

### 10.9.1. La narración en el pasado

Tomamos como punto de partida la pareja de tiempos correspondientes a la narración y a la descripción, que son los dos tiempos centrales de la historia.

Se emplea el pretérito indefinido y señala **acciones acabadas**:

#### Ejemplo de secuencia en pasado:

Fue al servicio, miró a la ventana y vio la hora. Se dio cuenta que se había quedado dormido.

Si queremos describir lo que le ocurre al personaje, para ello utilizaremos el pretérito imperfecto. Las expresiones **coinciden** con el tiempo de acción que se desarrolla en la narración:

#### Ejemplo de secuencia en pasado:

Se asomó y vio como se desarrollaba la acción fuera. Vio como se movían los árboles y se llevaban las hojas del suelo.

El pretérito pluscuamperfecto describe algo que sucede en el pasado y que no sucede **inmediatamente**. Lo vamos a ver ilustrado con otro ejemplo:

#### Ejemplo de secuencia en pasado:

Esa noche había llorado mucho. Se arrepentía de lo que había hecho la noche anterior.

El pretérito anterior se usa para narraciones que ocurren **inmediatamente**. Vamos a ver un ejemplo:

#### Ejemplo de secuencia en pasado:

Se levantó y preparó el desayuno cuando hubo dado de comer al perro.

Por último, vamos a introducir el pluscuamperfecto y se utiliza cuando el tiempo de la narración es cercano. Para tenerlo más claro, vamos a poner un ejemplo:

#### Frase mal construida en pasado:

Hacía un mes que se cayó en la zanja y no pudo recuperar su examen.

#### Frase bien construida en pasado:

Hacía un mes que se había caído en la zanja y no había podido recuperar el examen.

Si queremos narrar una acción posterior a la acción pasada vamos a utilizar el condicional simple, pero esta acción **no ha empezado**.



**Ejemplo narración condicional:**

Antes de iniciar la conversación le dijo que se ducharía cuando terminara de hacer ejercicio.

Por otro lado, tenemos el condicional compuesto que coincide con el simple, cuya acción ha sido **acabada**:

**Ejemplo narración condicional:**

Salió a toda prisa de casa para llegar a clase, pero se dió cuenta que las clases habrían empezado.

### 10.9.2. La narración en presente

Para narrar una acción que está sucediendo en ese mismo momento se utiliza el presente de indicativo. Y para ello, vamos a poner un ejemplo:

**Ejemplo de secuencia en presente:**

Se peina el pelo y se arregla para su próxima salida.

Para narrar una acción que se está realizando vamos a utilizar el presente simple. Para ver más clara esta acción vamos a poner un ejemplo:

**Ejemplo de secuencia en presente:**

Condujo a alta velocidad cuando una moto acelera a su lado.

Para narrar acciones que tienen lugar en un tiempo anterior en la narración se utiliza el pretérito indefinido. Para tenerlo más claro, vamos a poner un ejemplo:

**Ejemplo de secuencia en presente:**

La cena terminó tarde, por lo que le tocó coger el transporte público.

Por otro lado, para narrar acciones que son inmediatamente anteriores se utiliza el pretérito perfecto. Con su siguiente ejemplo:

**Ejemplo de secuencia en presente:**

Se prepara para ir al colegio cuando se da cuenta que se le ha olvidado un libro.

Y finalmente, para narrar una acción futura utilizaremos el futuro. El tiempo futuro lo dividiremos en simple si la acción no se ha terminado y en compuesto si la acción está terminada. Para tenerlo más claro vamos a poner un ejemplo:

**Ejemplo de acciones futuras:**

Cuando acabe el examen, decidirá donde cenar. Pero no sabe si habrá acabado para la hora de cenar.

### 10.9.3. Saltos temporales

Para desarrollar una buena novela hay que tener muy en cuenta los saltos temporales y es muy importante situar la acción en un tiempo.

### 10.10. Cursiva

#### Primera Regla

No se escriben en cursiva los nombres de barcos ni de mascotas.

#### Segunda Regla

Los extranjerismos sin adaptar se escriben en cursiva.

Los extranjerismos pueden ser adaptados o inadaptados. Los adaptados han sido modificados para que sean acordes al castellano, por lo que se escriben en letra redonda.

#### Ejemplos de extranjerismos adaptados:

Fútbol  
Carné

Por otro lado, los inadaptados se escriben en cursiva.

#### Ejemplos de extranjerismos inadaptados:

*Pizza*  
*Software*

### 10.11. Mayúsculas

#### Primera Regla

Las mayúsculas siempre llevan tilde.

#### Segunda Regla

Los puntos cardinales ya no se escriben en mayúsculas.

#### Ejemplo de puntos cardinales:

El barco fue al norte.  
La borrasca se dirigió al sureste.

**Excepción:** Los nombres de los Polos se escriben en mayúsculas.

#### Tercera Regla

Los títulos y los cargos no se escriben en mayúsculas.

**Cuarta Regla**

Tierra se escribe en mayúscula; sol y luna, en minúsculas.

**Quinta Regla**

Los días de la semana y los meses van siempre en minúscula.

**Sexta Regla**

La Muerte, el Bien y el Mal.

## 10.12. Laísmo, leísmo y loísmo

### 10.12.1. Laísmo

El laísmo consiste en usar la y las para señalar el complemento indirecto femenino. Es la sustitución incorrecta de los pronombres de complemento indirecto le y les.

**Malos usos:**

La di un beso.  
Dilas que vengan.  
No las habla desde hace meses.

**Buenos usos:**

Le di un beso.  
Diles que vengan.  
No les habla desde hace meses.

### 10.12.2. Loísmo

El loísmo consiste en usar lo y los para señalar el complemento indirecto masculino. Es la sustitución incorrecta de los pronombres de complemento indirecto le y les.

**Malos usos:**

Lo diseñó con mucha dedicación.  
Los pidió la semana pasada.  
A mis primos los multaron.

**Buenos usos:**

Le diseñó con mucha dedicación.  
Les pidió la semana pasada.  
A mis primos les multaron.

---

### 10.12.3. Leísmo

El leísmo consiste en usar los pronombres le y les en lugar de lo, los, la y las.

**Malos usos:**

El balón le cogí y me fui.  
La casa le compró muy barata.

**Buenos usos:**

El balón lo cogí y me fui.  
La casa la compré muy barata.

### 10.13. Gerundio

Regla Esencial

La acción debe ser inmediata o simultánea.

Primera Regla

La acción ocurre al mismo tiempo que la acción principal.

Segunda Regla

La acción del gerundio ocurre inmediatamente antes que la acción principal.

Tercera Regla

La acción del gerundio ocurre inmediatamente después de la acción principal.

### 10.14. Los títulos

Primera Regla

Escribiremos en mayúscula la primera palabra de un título.

**Primera excepción:** Si el título es un nombre propio se escribirá en mayúscula.

**Segunda excepción:** Se escriben en mayúsculas todas las palabras significativas.

Segunda Regla

Se puede saltar la primera regla si el título de la obra está en otro idioma.

Tercera Regla

Los títulos se escriben en cursiva.

**Primera excepción:** El texto sea en cursiva, por lo que se tendrá que escribir en redonda.

**Segunda excepción:** Si mencionamos un disco o un tema se escribirá en letra redonda

y entre comillas.

**Tercera excepción:** Los títulos de los libros sagrados se escriben en redonda.

**Cuarta excepción:** En un texto a mano la cursiva de los títulos se escriben entre comillas.

#### Cuarta Regla

Si el título va en cursiva, lo tendremos que escribir entre comillas.

#### Quinta Regla

Nunca se escribe punto después de los títulos y los subtítulos de libros.

---



# Capítulo 11

## Guión para TFGs

### 11.1. Introducción

Un Trabajo de Fin de Grado (TFG) es una asignatura esencial para poder superar la carrera y obtener el título. A la hora de seleccionarlo podemos encontrarnos dos maneras, el alumno puede seleccionar entre los distintos trabajos propuestos por los profesores o proponerle el trabajo a un profesor.

Tiene mucha duración, ya que, cuesta los créditos que cuesta.

Lo primero que tenemos que tener en cuenta a la hora de realizar el trabajo es la organización, ya que, el alumno va a tener una serie de exámenes y trabajos de otras asignaturas. Por lo que tendrá que hacer una primera organización del trabajo lo antes posible.

Y finalmente, una vez el alumno se ha organizado, empezará a dar título a los distintos apartados del trabajo, se hará con el uso de las tecnologías aplicadas y terminará de escribir la memoria.

#### 11.1.1. Objetivos

En este apartado vamos a ver los puntos más importantes de las fases por los que tiene que pasar un TFG:

- a). Metodología de trabajo.
- b). Redacción del trabajo.
- c). Realización de presentación y defensa del trabajo delante del tribunal de evaluación.

#### 11.1.2. Metodología de trabajo

El TFG va a presentar varias fases durante su desarrollo:

- Estado del arte.
- Desarrollo de los contenidos.
- Redacción de la memoria.

Una vez que se ha empezado a desarrollar el trabajo es muy fácil liarse con versiones intermedias. Por lo que es importante, antes de empezar, organizar y estructurar toda la información.

Desde un punto de vista general, el desarrollo del TFG es prioritario y la memoria se escribe después. Es muy importante desarrollar muy bien la memoria antes de comenzar a escribir cualquier línea de código. Una vez el alumno quiera comenzar a desarrollar el TFG, podrá descargar una plantilla de L<sup>A</sup>T<sub>E</sub>X de TFG en la que podrá poner todo el contenido.

Es muy importante desarrollar la memoria a la vez que se desarrolla el TFG por los siguientes puntos:

- Presentación instantánea. Se podrá realizar una presentación, ya que, la memoria estará acabada.
- Es muy importante una buena planificación del tiempo para que en la fase de escritura no haya repeticiones ni contenido repetido.
- No realizar el trabajo al final para no tener que volver a repetir fases.
- No dejar retrasada la tarea:
  - Desaparecerá la motivación para realizar el trabajo.
  - No se entiende la importancia del trabajo.
  - Habrá estrés en el alumno si tiene dificultades para redactar.
- Hay que organizar todo el trabajo, para evitar errores y poder realizar un texto formal.
- Detectar los fallos lo antes posible para poder enmedarlos.
- No perder tiempo en explicar el texto.

## Ordenador

Es un medio portátil donde el alumno va a llevar toda la información de su trabajo por todo lo que es en si.

## Organización y seguridad

Durante el desarrollo del TFG, el alumno va a mirar muchas fuentes, códigos y distintas versiones del programa.

Es importante que el alumno tenga un directorio del disco duro exclusivo para el TFG. Y dentro de ese directorio, colocaremos todo el material que se vaya recopilando de las páginas web y según se vaya obteniendo más información, el alumno creará subcarpetas. Es importante que el alumno tenga en cuenta que toda la información que no le sirva pueda ser borrada para poder aclarar las ideas y deberá proteger la información de malware y de fallos del ordenador. No solo el alumno tiene que hacer todo lo anterior, sino que además, por seguridad deberá realizar lo siguiente:

- Realizar una copia del TFG en una nube tipo Google Drive o varias similares.
- Realizar una copia del TFG en un disco duro.



- Realizar una copia del TFG en un medio físico externo.
- Realizar copias de seguridad con frecuencia.
- Consolidar hitos parciales.

## 11.2. Formato

Para realizar un TFG extenso se le recomienda utilizar al alumno aplicaciones más profesionales como L<sup>A</sup>T<sub>E</sub>X. Dentro de este, en algunas de estas aplicaciones se pueden crear entornos más familiares para el alumno para que pueda confeccionar su TFG con tranquilidad.

El formato de entrega suele ser tanto en físico como en electrónico.

Es importante que el alumno lea todos los requisitos de la memoria antes de confeccionar el TFG, ya que, puede haber problemas en el formato que no se detecten antes de la impresión.

### 11.2.1. Lengua

La lengua en la que se realice la memoria será especificada por la universidad.

### 11.2.2. Imágenes

Es muy importante que el alumno no ponga más imágenes de las que sean necesarias, y además, que sepa que las imágenes que sean puestas tendrán un texto acorde a ellas.

### 11.2.3. Estilo

La redacción debe ser realizada de una forma activa y de forma impersonal, por lo tanto, el alumno deberá emplear los verbos en infinitivo o en tercera persona del singular o plural reflexivo. Por lo tanto, deberá evitar usar tiempos como la primera persona del singular o del plural.

También se va a valorar de una manera muy positiva que el alumno tenga un amplio vocabulario y que se vea que tiene conocimiento de lo que escribe.

Otra cosa importante a la hora de la redacción de la memoria es que el alumno debe evitar a toda costa utilizar un lenguaje coloquial, ya que, su trabajo lo van a leer muchas personas distintas y tribunales distintos.

Y finalmente, el alumno tiene que saber que en su trabajo tiene todo lo necesario para obtener su título. Por lo que, todos los lectores que lean el trabajo del alumno se darán cuenta de que el trabajo ha sido realizado con su esfuerzo y dedicación.

## 11.3. Estructura

El trabajo tiene que estar organizado de una manera predefinida propuesta por los tribunales de la universidad. Esto facilita la estructuración del trabajo y la comprensión por parte del alumno y del tribunal.

La memoria va a estar dividida en los siguientes puntos expuestos a continuación.

---

### **11.3.1. Preliminares**

En esta parte se expone la información general del trabajo y las partes que el alumno no ha querido incluir en el mismo.

Aquí las páginas no deberían estar numeradas y si lo estuvieran, deberían estar numeradas en números romanos.

### **11.3.2. Título**

Es obligatorio y es lo que da carácter al trabajo. Se recomienda que sea sencillo y que mediante él, se entienda la obra.

Se recomienda que sean menos de 10 palabras y que no contenga ni abreviaturas ni acrónimos.

### **11.3.3. Subtítulo**

Es optativo y aporta una información extra sobre el trabajo.

### **11.3.4. Autor/a y año de creación de la obra**

Es obligatorio y debe incluir el nombre completo del autor. El año que se expone es el año del depósito del trabajo.

### **11.3.5. Director/a y departamento al que está adscrito el tutor del TFG**

Es optativo y depende de la escuela o facultad.

### **11.3.6. Dedicatoria**

Depende de lo que quiera el alumno. El orden de los agradecimientos es importante, por ello, es importante que el alumno repase el orden en el trabajo definitivo.

### **11.3.7. Agradecimientos**

Es optativo y se puede colocar al principio o al final.

En esta parte se agradece el trabajo de el compañero de trabajo y del tutor.

### **11.3.8. Otras partes**

Las otras partes aquí expuestas dependen de la normativa de cada facultad o escuela.

### **11.3.9. Resumen**

Es mejor redactarlo al final, ya que, es la síntesis de todo el trabajo expuesto y en el deben aparecer los puntos más importantes expuestos en el trabajo. Se puede describir de una manera concisa, las herramientas y la metodología.

La extensión no debe superar las 200 o 300 palabras. Y es importante utilizar palabras clave.

## Palabras clave

Son para tipificar y clasificar el trabajo en la base de datos. El alumno utilizará de 3 a 10 y pueden ser compuestas.

## Prefacio o prólogo

Destaca los méritos, el valor del trabajo y da contexto a la obra. Puede ser opcional.

### 11.3.10. Índices y glosarios

Su objetivo es localizar la información relevante.

Los índices van al principio, antes de la introducción y después de los agradecimientos.

### 11.3.11. Índice o sumario

No tiene que ser muy extenso en nivel de detalle y tiene que estar estructurado. Es obligatorio.

### 11.3.12. Índices generales, tablas e imágenes

Suelen ser opcionales.

**Índice o glosario:** Lista ordenada de conceptos y nombres propios. Es importante que aparezca al final del capítulo.

**Índice de imágenes:** Lista de tablas ordenadas a lo largo de toda la memoria. Hace referencia a las imágenes aparecidas en la memoria.

**Abreviaturas:** Depende del desarrollo de trabajo utilizarlas o no. Se puede colocar como apéndice.

### 11.3.13. Otras

Es importante que se demuestre la autenticidad del trabajo para evitar complicaciones con el tribunal o con otros lectores del trabajo.

### 11.3.14. Cuerpo principal

Los puntos que deberían aparecer en el cuerpo son la introducción, el estado del arte, análisis del problema, y muchos otros puntos.

## Introducción

Esta parte es la que va a dar el contexto a la obra. En este apartado se van a justificar los puntos más importantes del trabajo.

El trabajo debe ser presentado de una manera simple, de tal forma, que personas ajenas a la materia sean capaces de entenderlo.

Puede ayudar a la introducción que aparezca un glosario al final.

Y aquí encontraremos los siguientes puntos.

---

## Motivación

Es muy importante en este punto exponer los motivos y la temática del trabajo. Y si se han mostrado algunas alternativas, también deberían ser expuestas.

## Objetivos

En este apartado se van exponer de una manera clara y concisa los puntos más importantes del trabajo. Aquí el autor mostrará la intencionalidad del trabajo. Se pueden plantear todos los objetivos y los siguientes subobjetivos que han ido saliendo a lo largo del trabajo. Los puntos expuestos deben ser:

- Concretos.
- Factibles.
- Identificar el tema, el destino del trabajo y el uso de los recursos.
- Verbos en infinitivo.

## Metodología

En esta parte se exponen los pasos que se han realizado para llegar a los objetivos. Se debe describir con todo detalle y guiará a quien lo lea, por lo que todos los métodos utilizados deben ser descritos con el mayor detalle posible.

## Estructura de la obra

Es un párrafo grande que resume el contenido, justifica y presenta la estructura general de la obra para saber a dónde saltar según los intereses del lector o del alumno. Está marcada por la metodología y debe ser coherente con ella.

## Convenciones

Los convencionalismos se utilizan para alguna normativa, por ejemplo:

- El código fuente se muestra en letra courier cursiva.
- Las palabras extranjeras se marcarán en cursiva.
- Se entrecomillan las citas textuales.

### 11.3.15. Estado del arte

En este apartado se muestran los puntos fuertes de la tecnología de su utilidad, se muestran en qué ámbitos se aplican y cómo puede ser utilizada.

Si vamos a mostrar como funciona una tecnología en formato vídeo es muy importante reseñarlo con una URL que de acceso al vídeo. Y relacionar esta URL en la referencia bibliográfica.

También es muy importante tener en este punto tener un amplio de referencias bibliográficas que justifiquen cada uno de los puntos expuestos en el estado.

### **Crítica al estado del arte**

En este punto se detecta cualquier fallo o laguna expuesto en el trabajo.

### **Propuesta**

Justifica los puntos expuestos en la crítica.

### **11.3.16. Análisis del problema**

En este apartado se sistematiza un buen análisis de requisitos.

#### **Análisis de requisitos**

Son los requisitos que se le van a pedir a la aplicación del TFG.

#### **Análisis de las soluciones**

Este es el apartado fundamental del trabajo. Aquí se van a plantear soluciones para los requisitos en el apartado anterior. Habrá que presentar una serie de soluciones con sus ventajas y desventajas.

#### **Solución propuesta**

En este apartado se va a desarrollar la solución elegida y en esta parte se va a exponer los diagramas de estado, algoritmos en pseudocódigo,...

#### **Análisis de seguridad**

Este apartado no lo tienen porque tener todos los trabajos. Pero es importante exponerlo, ya que, es un problema de que el alumno deba ser consciente.

#### **Análisis energético o de eficiencia algorítmica**

Este apartado hace referencia a la batería de los objetivos móviles. Se deberían realizar algoritmos eficientes para la aplicación, para reducir el consumo eléctrico. Además, el alumno debe especificar los detalles que le preocupen en este apartado.

#### **Análisis de la protección de datos**

En este apartado, se especificará el tipo de acceso a la información por parte de los usuarios. No todos los trabajos tienen porque tenerlo.

#### **Análisis de la internacionalización**

Es opcional y consiste en desarrollar una API interna que sólo utilizarán los programadores.

---

**Presupuesto**

En este apartado se propone realizar una relación de los materiales que ha utilizado el alumno y su esfuerzo.

Para realizarlo, el alumno realizará un diagrama de Gantt para desarrollar las fases del proyecto y el número de horas empleadas.

**Colaboración**

Este apartado es importante, en él se incluyen las personas implicadas y todos los proyectos que son ajenos al mismo. Es un apartado obligatorio.

**11.3.17. Diseño de la solución**

En este apartado se determinan las herramientas que implementan las soluciones.

**Análisis de las herramientas**

Es un apartado importante, ya que, se debe decidir el entorno del desarrollo, el lenguaje a utilizar, las APIs...

**Arquitectura del software**

En este apartado se describirán los distintos elementos que influyen en la ejecución del programa.

**11.3.18. Implementación**

Es importante no incluir el código fuente dentro del cuerpo de la memoria ni en los apéndices.

Solo se puede incluir el código cuando:

- a). Cuando puede influir en la fase de análisis.
- b). Cuando la implementación sea difícil de entender.

El código normalmente se presentará como regla general:

- a). En los párrafos adjuntos de la memoria.
- b). En los apéndices.

**11.3.19. Resultados**

En este apartado se muestran las pruebas que verifican la solución.

También se debe mostrar las pruebas de carga para su eficiencia.

## Discusión

En este apartado se analiza y verifica los resultados presentados anteriormente. Es la parte esencial del trabajo, pero es la parte más compleja de elaborar y organizar. Como sugerencia para realizarla se debería:

- Responder a las preguntas de la introducción.
- Escribir los resultados como evidencia científica.
- Informar de los resultados tanto de los positivos como de los negativos.
- Especular y teorizar con imaginación y lógica.
- Incluir recomendaciones.
- Evitar sacar más conclusiones de las que los resultados permitan.

### 11.3.20. Conclusiones

En este apartado se da una respuesta al problema que originó el trabajo. Debe ser redactado de una forma sencilla, coherente y que en él se respondan todos los problemas expuestos en la introducción. Es importante relacionar bien los objetivos con su solución y seguir toda la línea de desarrollo del trabajo.

Por otro lado, es muy importante no repetir lo dicho en la conclusión en los resultados y no explicar más de los objetivos de los planteados. Aquí el alumno se tiene que plantear:

- Los errores cometidos durante su desarrollo.
- Lo aprendido durante el desarrollo del proyecto.
- La capacidad de aprendizaje y la iniciativa del alumno.
- Las carencias de su trabajo.
- Las consecuencias de su trabajo en la realidad.

### Relación trabajo con estudios cursados

No es un apartado obligatorio, pero es un apartado interesante, ya que, el alumno muestra todo lo aprendido y su relación en el desarrollo del trabajo.

El objetivo de este apartado es coordinar los conocimientos recibidos durante la carrera y demostrar que se puede dar soluciones a los problemas reales. Es muy importante tener cuidado con:

- a). Utilizar pocas tecnologías que:
    - a) No se demuestra el dominio tecnológico del alumno.
    - b) El alumno sabe sobre las tecnologías de las que habla.
  - b). No utilizar muchas tecnologías para que el trabajo no sea muy largo y se emplee demasiado tiempo.
-

## Trabajos futuros

En este apartado se realiza una lista que detalla lo siguiente:

- Otros puntos que al alumno le hubiera gustado tratar.
- Otras líneas de desarrollo que se han encontrado durante el desarrollo del proyecto.
- Mejoras en las funcionalidades del trabajo.

Es importante indicar en este apartado cuales caminos no se han seguido y las razones por las que no se han seguido.

### 11.3.21. Bibliografía

Este es otro apartado esencial y permite comprobar:

- a). Verificar las fuentes originales.
- b). No ha plagiado frases de otros trabajos.
- c). Profundizar en el trabajo realizado.

No hace falta relacionar todos los conceptos, sino que relacionaremos los que sean necesarios. Otros puntos importantes son los siguientes:

- No hacer referencias a páginas web.
- Si se hace una referencia de una web de un fabricante o un producto se hará a pie de página.
- No puede aparecer ninguna referencia bibliográfica que no esté en el texto.
- Puede haber varias referencias en el texto a la misma fuente bibliográfica.

### 11.3.22. Referencias a pie de página

En este apartado se referencia los enlaces a las páginas web de las herramientas o del producto y se realizan a pie de página. Solo se referencia la primera vez.

### 11.3.23. Anexos

En este apartado se trata los aspectos no necesarios tratados en el trabajo.

Pone en valor la cantidad de trabajo realizado aunque no haya ido en la dirección del trabajo.



## 11.4. Tratamientos especiales

### 11.4.1. Código fuente

En este apartado, como regla general, se incluye que:

- Hay que poner la menor cantidad de código fuente en la memoria.
- Extensión justa del código fuente.
- Poner el código en el cuerpo de la obra, sino colocar en un apéndice.
- Comentarlo todo.
- Justificar la implementación del código.
- Resaltar el código y justificarlo.

### Correcciones ortográficas y de estilo

Es importante que la memoria no contenga faltas de ortografía. Si se está utilizando el procesador de textos, se recomienda activar el corrector. También el alumno debe evitar usar frases largas y lenguaje difícil de entender.

### 11.4.2. Defensa pública del TFG

El alumno tiene 20 minutos para defender su trabajo.

Es fundamental que el acto de comunicación se centre no sólo en la demostración de la capacidad técnica del alumno y en los contenidos técnicos, sino también se demuestre las habilidades comunicativas del alumno.

### 11.4.3. Diapositivas

Esta parte es esencial en la presentación del trabajo, la diapositiva contendrá la información justa en poco texto. Debido al tiempo empleado en la exposición, el número de diapositivas no excederá las 15.

### Público objetivo

El alumno tiene que tener en cuenta que no es el público objetivo:

- La sala o la audiencia.
- El mismo locutor.

El público objetivo del alumno es el tribunal:

- 3 y 5 personas.
- Profesores que imparten la titulación.
- Especialistas en la materia.

Hay que evitar a toda costa las diapositivas básicas.

---

#### **11.4.4. Comunicador**

El alumno debe comunicar todo lo destacable de la diapositiva, por eso, es importante que la diapositiva contenga la información justa y que alguien que solo lea las diapositivas no entienda el trabajo.

#### **11.4.5. Estructura de los contenidos**

Es importante que el contenido de las diapositivas esté estructurado de la siguiente forma:

- a). Presentación (1 diapositiva).
- b). Introducción (1 diapositiva).
- c). Estado del arte (1-2 diapositivas).
- d). Crítica al estado del arte (1 diapositiva).
- e). Solución al trabajo realizado (1 diapositiva).
- f). Diseño, arquitectura, esquema de la solución realizada (3 diapositivas).
- g). Resultados (1 diapositiva).
- h). Conclusiones (1 diapositiva).

#### **11.4.6. El día de la presentación**

Es muy importante tomarse en serio la presentación del trabajo, una comparación sería con la presentación del proyecto a una empresa.

#### **11.4.7. Recomendaciones**

Es muy importante llegar con tiempo de antelación a la presentación para evitar errores, además el alumno probará la presentación con antelación.

#### **11.4.8. Malas prácticas**

Es importante destacar a que no se espere a subir al estrado para probar la presentación, el alumno deberá mostrar tranquilidad durante la presentación y sobre todo, deberá evitar improvisar.

### **11.5. Manejo de ideas**

En PL/SQL una advertencia o condición de error es llamada una excepción. Estas pueden ser definidas en forma interna (en tiempo de ejecución de un programa) o explícitamente por el usuario. Ejemplos de excepciones definidas en forma interna son la división por cero y la falta de memoria en tiempo de ejecución. Estas mismas condiciones excepcionales tienen sus nombres propios y pueden ser referenciadas con ellos: `zero_divide` y `storage_error`. También se pueden definir excepciones a medida y nombrarlas de alguna forma, utilizando las reglas de construcción mencionadas en el capítulo 2.

Cuando ocurre un error se alcanza la excepción, esto quiere decir que se ejecuta la porción del programa donde ésta se encuentra implementada, transfiriéndose el control a ese bloque de sentencias. Las excepciones definidas por el usuario deben ser alcanzadas explícitamente utilizando la sentencia raise.

Con las excepciones se pueden manejar los errores cómodamente sin necesidad de mantener múltiples chequeos por cada sentencia escrita. También provee claridad en el código desde el momento en que permite mantener las rutinas correspondientes al tratamiento de los errores en forma separada de la lógica del negocio

## 11.6. Excepciones predefinidas

Las excepciones predefinidas no necesitan ser declaradas. Simplemente se utilizan cuando estas son gatilladas por algún error determinado.

La siguiente es la lista de las excepciones predeterminadas por PL/SQL y una breve descripción de cuándo son accionadas:

[HTML]9B9B9B[HTML]C126BD	[HTML]C126BD	[HTML]C126BD
[HTML]9B9B9B		
[HTML]9B9B9BACCESS INTO NULL	El programa intentó asignar valores a los atributos de un objeto no inicializado	-6530
[HTML]9B9B9BCOLLECTION IS NULL	El programa intentó asignar valores a una tabla anidada aún no inicializada	-6531
[HTML]9B9B9BCURSOR ALREADY OPEN	El programa intentó abrir un cursor que ya se encontraba abierto.	
[HTML]9B9B9B	Recuerde que un cursor de ciclo FOR automáticamente lo abre y ello no se debe especificar con la sentencia OPEN	-6511
[HTML]9B9B9BDUP_VAL_ON_INDEX	El programa intentó almacenar valores duplicados en una columna que se mantiene con restricción de integridad de un índice único (unique index)	-1
[HTML]9B9B9B		
[HTML]9B9B9BINVALID CURSOR	El programa intentó efectuar una operación no válida sobre un cursor	-1001
[HTML]9B9B9BINVALID_NUMBER	En una sentencia SQL, la conversión de una cadena de caracteres hacia un número falla cuando esa cadena no representa un número válido	-1722
[HTML]9B9B9B		
[HTML]9B9B9BLOGIN DENIED	El programa intentó conectarse a Oracle con un nombre de usuario o password inválido	-1017
[HTML]9B9B9BNO_DATA_FOUND	Una sentencia SELECT INTO no devolvió valores o el programa referenció un elemento no inicializado en una tabla indexada	+100
[HTML]9B9B9B		
[HTML]9B9B9BNOT_LOGGED_ON	El programa efectuó una llamada a Oracle sin estar conectado	-1012
[HTML]9B9B9BPROGRAM_ERROR	PL/SQL tiene un problema interno	-6501
[HTML]9B9B9BROWTYPE_MISMATCH	Los elementos de una asignación (el valor a asignar y la variable que lo contendrá) tienen tipos incompatibles. También se presenta este error cuando un parámetro pasado a un subprograma no es del tipo esperado	-6504
[HTML]9B9B9B		
[HTML]9B9B9BSELF IS NULL	El parámetro SELF (el primero que es pasado a un método MEMBER) es nulo	-30625
[HTML]9B9B9B		
[HTML]9B9B9BSTORAGE_ERROR	La memoria se terminó o está corrupta	-6500
[HTML]9B9B9BSUBSCRIPT_BEYOND_COUNT	El programa está tratando de referenciar un elemento de un arreglo indexado que se encuentra en una posición más grande que el número real de elementos de la colección	-6533
[HTML]9B9B9B		
[HTML]9B9B9BSUBSCRIPT_OUTSIDE_LIMIT	El programa está referenciando un elemento de un arreglo utilizando un número fuera del rango permitido (por ejemplo, el elemento "-1")	-6532
[HTML]9B9B9B		
[HTML]9B9B9BSYS_INVALID_ROWID	La conversión de una cadena de caracteres hacia un tipo rowid falló porque la cadena no representa un número	-1410
[HTML]9B9B9BTIMEOUT_ON_RESOURCE	Se excedió el tiempo máximo de espera por un recurso en Oracle	-51
[HTML]9B9B9BTOO_MANY_ROWS	Una sentencia SELECT INTO devuelve más de una fila	-1422
[HTML]9B9B9BVALUE_ERROR	Ocurrió un error aritmético, de conversión o truncamiento. Por ejemplo, sucede cuando se intenta calzar un valor muy grande dentro de una variable más pequeña	-6502
[HTML]9B9B9B		
[HTML]9B9B9BZERO_DIVIDE	El programa intentó efectuar una división por cero	-1476

### 11.6.1. Excepciones definidas por el usuario

PL/SQL permite al usuario definir sus propias excepciones, las que deberán ser declaradas y gatilladas explícitamente utilizando otros comandos del lenguaje.

#### Declaración

Las excepciones sólo pueden ser declaradas en el segmento “Declare” de un bloque, subprograma o paquete. Se declara una excepción escribiendo su nombre seguida de la palabra clave EXCEPTION. Las declaraciones son similares a las de variables, pero recuerda que una excepción es una condición de error, no un ítem de datos. Aun así, las mismas reglas de alcance aplican tanto sobre variables como sobre las excepciones.

#### Ejemplo:

```

1 DECLARE
2     error_01     EXCEPTION;
```

#### Reglas de Alcance

Una excepción no puede ser declarada dos veces en un mismo bloque. Tal como las variables, una excepción declarada en un bloque es local a ese bloque y global a todos los

sub-bloques que comprende.

La sentencia “RAISE”

La sentencia RAISE permite gatillar una excepción en forma explícita. Es factible utilizar esta sentencia en cualquier lugar que se encuentre dentro del alcance de la excepción.

**Ejemplo:**

```
1 DECLARE
2     out_of_stock     EXCEPTION;           -- declaración de la
3     total            NUMBER(4);         excepción
4 BEGIN...
5
6     IF total < 1 THEN
7         RAISE out_of_stock;           -- llamado a la excepción
8     END IF;
9 EXCEPTION
10    WHEN out_of_stock THEN
11        -- manejar el error aquí
12    WHEN OTHERS THEN...
13
14 END;
```

Finalmente, cabe destacar la existencia de la excepción OTHERS, que simboliza cualquier condición de excepción que no ha sido declarada. Se utiliza comúnmente al final del bloque de excepciones para absorber cualquier tipo de error que no ha sido previsto por el programador. En ese caso, es común observar la sentencia ROLLBACK en el grupo de sentencias de la excepción o alguna de las funciones SQLCODE – SQLERRM, que se detallan en el próximo punto.

### 11.6.2. Uso de SQLCODE y SQLERRM

Al manejar una excepción es posible apoyarse con las funciones predefinidas SQLCode y SQLErrm para aclarar al usuario la situación de error acontecida.

Sqlcode siempre retornará el número del error de Oracle y un “0” (cero) en caso exitoso al ejecutarse una sentencia SQL.

Por otra parte, Sqlerrm retornará el correspondiente mensaje de error para la situación ocurrida. También es posible entregarle a la función SQLERRM un número negativo que represente un error de Oracle y ésta devolverá el mensaje asociado.

Estas funciones son muy útiles cuando se utilizan en el bloque de excepciones, para aclarar el significado de la excepción OTHERS, cuando ésta ocurre.

Estas funciones no pueden ser utilizadas directamente en una sentencia SQL, pero sí se puede asignar su valor a alguna variable de programa y luego usar esta última en alguna sentencia.

**Ejemplo:**

```
1 DECLARE
2     err_num          NUMBER;
3     err_msg          VARCHAR2(100);
```

```
4 ...
5 BEGIN
6 EXCEPTION
7     WHEN OTHERS THEN
8         err_num := SQLCODE;
9         err_msg := SUBSTR(SQLERRM, 1, 100);
10        INSERT INTO errores VALUES(err_num, err_msg);
11 END;
```

---



# Índice de figuras

1.1.	Logo Overleaf . . . . .	1
1.2.	Logo ShareLatex . . . . .	1
1.3.	Ejemplo de artículo . . . . .	3
1.4.	Ejemplo de un libro . . . . .	6
1.5.	Ejemplo de presentación . . . . .	9
1.6.	Página inicio Overleaf . . . . .	10
1.7.	Menú Overleaf . . . . .	11
1.8.	Crear nuevo proyecto . . . . .	12
1.9.	Salto de página . . . . .	17
1.10.	Apartado, subapartado y título . . . . .	21
1.11.	Error código . . . . .	25
2.1.	Captura página web . . . . .	32
2.2.	Ejemplo de tabla . . . . .	32
2.3.	Generando tabla y copiándola . . . . .	33
2.4.	Modificación tabla . . . . .	33
2.5.	Crear una nueva tabla . . . . .	34
2.6.	Filas y columnas . . . . .	34
2.7.	Imagen de ejemplo . . . . .	35
2.8.	Página selección color . . . . .	37
3.1.	Ejemplo de texto en multicol . . . . .	40
3.2.	Ejemplo de texto en multicol con separación . . . . .	41
3.3.	Ejemplo de multicol sin formato estándar . . . . .	42
3.4.	Ejemplo de multicol con líneas verticales azules . . . . .	43
3.5.	Documento con licencia . . . . .	63
3.6.	Encabezado y pie de página . . . . .	65
4.1.	Itemizados personalizados . . . . .	69
4.2.	Tabla para utilizar en el ding . . . . .	71
4.3.	Ejemplo de itemizado con ding . . . . .	71
4.4.	Letra en un enumerado . . . . .	72
6.1.	Beamer básico . . . . .	87
6.2.	Bloque básico . . . . .	89
6.3.	Bloque alerta básico . . . . .	89
6.4.	Bloque ejemplo básico . . . . .	90
6.5.	Bloque color amarillo . . . . .	91
6.6.	Comando pause . . . . .	92

---

6.7. Comando pause . . . . .	93
6.8. Comando onslide . . . . .	95
6.9. Comando only . . . . .	96
6.10. Comando invisible . . . . .	97
6.11. Comando visible . . . . .	99
6.12. Presentación tipo Madrid . . . . .	101
6.13. Presentación tipo Antibes . . . . .	102
6.14. Presentación tipo Hannover . . . . .	103
6.15. Presentación tipo Berlin . . . . .	104
6.16. Presentación tipo Malmoe . . . . .	105
7.1. Gráfica de barras . . . . .	110
7.2. Diagrama de círculos . . . . .	111
7.3. Fondo cambiado de color . . . . .	115
9.1. Ejemplo de bibliografía . . . . .	126
9.2. Referencia citada . . . . .	127
9.3. Ejemplo BibTex . . . . .	129
9.4. Ejemplo BibTex 2 . . . . .	130
9.5. Colocación bibliografía . . . . .	131
9.6. Buscador Google Scholar . . . . .	132
9.7. Búsqueda en Google Scholar . . . . .	133
9.8. Citar Google Scholar . . . . .	133
9.9. Citación en BibTex de Google Scholar . . . . .	133



# Índice de cuadros

2.1. Tabla con nombre . . . . .	31
3.1. Letras griegas . . . . .	46
3.2. Letras griegas AMS . . . . .	46
3.3. Símbolos flechas . . . . .	46
3.4. Flechas AMS . . . . .	47
3.5. Flechas negación AMS . . . . .	47
3.6. Operadores binarios . . . . .	47
3.7. Operadores de tamaño variable . . . . .	47
3.8. Operadores de relación . . . . .	48
3.9. Signos de puntuación . . . . .	48
3.10. Otros símbolos . . . . .	48
10.1. Tipos de comillas . . . . .	139



# Listings

1.1. Código artículo . . . . .	3
1.2. Código libro . . . . .	6
1.3. Código presentación . . . . .	9
1.4. Código preámbulo . . . . .	12
1.5. Código preámbulo . . . . .	13
1.6. Código comandos . . . . .	14
1.7. Código entornos . . . . .	14
1.8. Código en negrita . . . . .	15
1.9. Código en redonda . . . . .	15
1.10. Código en itálica . . . . .	15
1.11. Código en máquina de escribir . . . . .	15
1.12. Código en máquina de versalita . . . . .	15
1.13. Código de texto centrado . . . . .	16
1.14. Código de texto a la izquierda y a la derecha . . . . .	16
1.15. Código de salto de línea y espaciado . . . . .	17
1.16. Código de salto de página . . . . .	17
1.17. Código de tamaño normal . . . . .	18
1.18. Código de tamaño diminuta . . . . .	18
1.19. Código de tamaño muy pequeño . . . . .	18
1.20. Código de tamaño más pequeño . . . . .	19
1.21. Código de tamaño pequeño . . . . .	19
1.22. Código de tamaño grande . . . . .	19
1.23. Código de tamaño más grande . . . . .	19
1.24. Código de tamaño muy grande . . . . .	19
1.25. Código de tamaño enorme . . . . .	20
1.26. Código de tamaño más enorme . . . . .	20
1.27. Código de tamaño negrita y gigante . . . . .	20
1.28. Código de ejemplo . . . . .	21
1.29. Ejercicio 1 . . . . .	22
1.30. Ejercicio 2 . . . . .	22
1.31. Ejercicio 3 . . . . .	23
1.32. Ejercicio 4 . . . . .	23
1.33. Ejercicio 5 . . . . .	24
1.34. Ejercicio 6 . . . . .	24
2.1. Ejemplo tabla básica . . . . .	29
2.2. Ejemplo tabla simple . . . . .	30
2.3. Ejemplo tabla simple con nombre . . . . .	31
2.4. Código inserción imágenes . . . . .	35
2.5. Colores 1 . . . . .	36

2.6. Colores personalizados . . . . .	37
3.1. Código landscape . . . . .	39
3.2. Código paquete multicolors . . . . .	40
3.3. Código paquete multicolors con separación . . . . .	41
3.4. Código paquete multicolors sin formato estándar . . . . .	42
3.5. Código paquete multicolors con líneas verticales azules . . . . .	43
3.6. Ejemplo de función matemática con el texto . . . . .	44
3.7. Ejemplo de función matemática con el texto resaltado . . . . .	44
3.8. Ejemplo de función matemática con el texto resaltado . . . . .	45
3.9. Ejemplo de función matemática con el texto resaltado . . . . .	45
3.10. Ejemplo de función matemática con el entorno de la ecuación sin numerar . . . . .	45
3.11. Ejemplo de función matemática con el entorno de la ecuación numerado . . . . .	45
3.12. Ejemplo del código de fórmulas simples . . . . .	48
3.13. Ejemplo del código de elevación de números . . . . .	49
3.14. Ejemplo del código de elevación de números . . . . .	49
3.15. Ejemplo del código de subíndice de números . . . . .	49
3.16. Ejemplo del código de subíndice de números . . . . .	50
3.17. Ejemplo del código de límites . . . . .	50
3.18. Ejemplo del código de límites . . . . .	51
3.19. Ejemplo del código de sumatorio . . . . .	51
3.20. Ejemplo del código de sumatorio . . . . .	51
3.21. Ejemplo del código de sumatorio . . . . .	51
3.22. Ejemplo del código de sumatorio con los subíndices cambiados . . . . .	52
3.23. Ejemplo del código de sumatorio con los subíndices a un lado . . . . .	52
3.24. Ejemplo del código de los dos comandos de fracciones . . . . .	53
3.25. Ejemplo del código de la fracción en los distintos entornos matemáticos . . . . .	53
3.26. Ejemplo del código del tamaño desigual de los delimitadores . . . . .	53
3.27. Ejemplo del código corrigiendo el tamaño de los delimitadores con left y right . . . . .	54
3.28. Ejemplo del código corrigiendo el tamaño de los delimitadores con Big . . . . .	54
3.29. Ejemplo del código de la diferencia entre los comandos de las raíces . . . . .	54
3.30. Ejemplo del código de una integral inmediata . . . . .	55
3.31. Ejemplo del código de una integral en dos puntos . . . . .	55
3.32. Ejemplo del código de las integrales dobles y triples . . . . .	56
3.33. Ejemplo del código de la integral junto con una fracción . . . . .	56
3.34. Ejemplo del código de la integral junto con una fracción con la integral bien dispuesta . . . . .	56
3.35. Ejemplo del código de la integral cerrada simple . . . . .	56
3.36. Ejemplo del código de una matriz . . . . .	57
3.37. Ejemplo del código de matriz entre corchetes . . . . .	57
3.38. Ejemplo del código de matriz entre paréntesis . . . . .	57
3.39. Ejemplo de código de operaciones entre matrices . . . . .	58
3.40. Ejemplo del código de un determinante . . . . .	59
3.41. Ejemplo del código de un texto dentro del modo matemático . . . . .	59
3.42. Ejemplo del código de brazo superior y brazo inferior . . . . .	59
3.43. Ejemplo del código de brazo superior y brazo inferior con texto . . . . .	60
3.44. Ejemplo del código de brazo superior y brazo inferior con texto y concatenados . . . . .	60
3.45. Ejemplo del código de sistemas de ecuaciones con llave a la izquierda . . . . .	61

3.46. Ejemplo del código de sistemas de ecuaciones con llave a la derecha . . . . .	61
3.47. Ejemplo del código de sistemas de ecuaciones sin llave . . . . .	62
3.48. Ejemplo del código del entorno eqnarray . . . . .	62
3.49. Ejemplo del código del entorno eqnarray con caja mbox . . . . .	63
3.50. Código licencia . . . . .	63
3.51. Código encabezado y pie de página . . . . .	65
3.52. Caja mdframed . . . . .	66
3.53. Tabla coloreada mdframed . . . . .	66
4.1. Código personalización itemize . . . . .	70
4.2. Código paquete pifont . . . . .	72
4.3. Código de enumerado con letra . . . . .	72
4.4. Código de pie de página . . . . .	74
4.5. Código de matriz y determinante . . . . .	74
5.1. Código tablero ajedrez . . . . .	77
5.2. Código empezar partida . . . . .	78
6.1. Código Beamer básico . . . . .	87
6.2. Código Beamer básico con include . . . . .	88
6.3. Código Beamer bloque básico . . . . .	89
6.4. Código Beamer bloque alerta básico . . . . .	89
6.5. Código Beamer bloque ejemplo básico . . . . .	90
6.6. Código Beamer bloque amarillo . . . . .	91
6.7. Código comando pause . . . . .	92
6.8. Código comando pause . . . . .	94
6.9. Código comando onslide . . . . .	95
6.10. Código comando only . . . . .	96
6.11. Código comando only . . . . .	98
6.12. Código comando visible . . . . .	99
6.13. Código comando estilo Madrid . . . . .	101
6.14. Código comando estilo Antibe . . . . .	103
6.15. Código comando estilo Hannover . . . . .	104
6.16. Código comando estilo Berlin . . . . .	104
6.17. Código comando estilo Malmoe . . . . .	105
7.1. Código diagrama básico . . . . .	110
7.2. Código diagrama círculos . . . . .	111
7.3. Código esquema simple . . . . .	112
7.4. Código esquema de llaves complejo . . . . .	112
7.5. Código diagrama de flujo . . . . .	113
7.6. Código cambio de nombre . . . . .	115
7.7. Código cambio de color de fondo . . . . .	116
8.1. Código verbatim . . . . .	117
8.2. Código verbatim texto . . . . .	117
regresionTFG.m . . . . .	119
8.3. Código de importar el código . . . . .	119
8.4. Código ejemplo personalizado . . . . .	120
8.5. Ejemplo con nombre . . . . .	121
8.6. Ejemplo con nombre código . . . . .	122
8.7. Código palabras clave . . . . .	122
9.1. Ejemplo de bibliografía . . . . .	126

9.2. Ejemplo de cita . . . . .	127
9.3. Ejemplo de BibTex . . . . .	129
9.4. Ejemplo de BibTex 2 . . . . .	130
9.5. Código bibliografía ejemplo 2 . . . . .	132

PASCAL

ABRIL 2019

Ult. actualización 25 de abril de 2019

TEX lic. LPPL & powered by **TEFLON** CC-ZERO

Este documento esta realizado bajo licencia Creative Commons “CC0 1.0 Universal”.

