

LATEX Y ALGUNAS DE SUS FUNCIONES

v. 4,3

Miguel Ángel García

8 de noviembre de 2021

Índice

1. Introducción	4
1.1. Creando un archivo	4
1.2. Preámbulo	4
1.2.1. Tipo de documento	4
1.2.2. Configuración del documento	5
1.3. Escritura en español (asientos y virgulilla)	6
1.3.1. Paquete <code>inputenc</code>	7
1.3.2. Paquete <code>babel</code>	7
1.4. Espacios	7
1.4.1. Espacios horizontales	7
1.4.2. Saltos verticales entre párrafos	8
1.4.3. Espacio entre párrafos con <code>setlength</code>	9
1.4.4. Medida de saltos verticales	10
1.5. Alineación del texto	11
1.5.1. Alineación con comando	11
1.5.2. Alineación con entornos	12
1.6. Letra	13
1.6.1. Tamaño de letra	13
1.6.2. Estilo de letra	14
1.6.3. Tipo de letra	14
1.6.4. Interlineado	15

2. Haciendo un taller	16
2.1. Portada	16
2.1.1. Portada con <code>maketitle</code>	16
2.1.2. Portada con entorno <i>titlepage</i>	17
2.2. Listas	19
2.2.1. Entorno <i>itemize</i>	19
2.2.2. Cambio de enumeración individual <i>itemize</i>	20
2.2.3. Entorno <i>enumerate</i>	21
2.2.4. Cambio de enumeración individual <i>enumerate</i>	22
2.2.5. Cambio de enumeración global	23
2.3. Citación y bibliografía	25
2.3.1. Archivo <code>bib.bib</code>	25
2.3.2. Referencias	27
2.3.3. Bibliografía	29
2.3.4. Sangría para citas (<code>quote</code>)	29
2.3.5. Paquete <code>hyperref</code>	29
2.4. Identación o sangría	29
2.4.1. Identación para párrafos	29
2.5. Formato secciones, subsecciones y subsubsecciones	29
2.5.1. <code>titleformat*</code>	30
2.5.2. <code>titleformat</code>	30
2.5.3. <code>titlespacing</code>	31
3. Tablas	34
3.1. Creación de tablas	34
3.2. Tamaño de columnas	36
3.3. Agrupación de filas	36
3.4. Agrupación de columnas	39
3.5. Color en las tablas	40
3.5.1. Color en las columnas	40
3.5.2. Color en las filas	41
3.5.3. Color en las celdas individuales	41
3.6. El paquete <code>array</code>	42
3.6.1. Definir tamaño de las columnas	42
3.6.2. Declaraciones en las columnas	43
3.7. Entorno <i>tabularx</i>	44
3.7.1. Tamaño de la tabla	45
3.7.2. Tabla a tamaño completo	46
3.8. Posicionar tablas (Tablas flotantes)	47
3.9. Entorno <i>longtable</i>	48

3.9.1.	Tablas sin encabezados y pies en todas las páginas	49
3.9.2.	Tablas con encabezados y pies en todas las páginas	50
3.10.	Otros	53
3.10.1.	División de celda de nombres de series	53
3.10.2.	Leyenda y citación	54
4.	Matemáticas	56
4.1.	Modo matemático en línea	56
4.2.	Modo matemático entre párrafos	56
4.2.1.	Entorno <code>equation</code>	57
4.3.	Escritura matemática	58
4.3.1.	Símbolos	58
4.3.2.	Espacios	58
4.4.	Paquete <code>amsmath</code>	59
4.4.1.	Entorno <code>equation*</code>	59
4.4.2.	Entorno <code>split</code>	60
4.4.3.	Entorno <code>multline</code>	61
4.4.4.	Entorno <code>align</code>	62
4.4.5.	Entorno <code>gather</code>	63
4.5.	Paquete <code>amssymb</code>	64
4.6.	Matrices	64
4.6.1.	Delimitantes personalizados	65
4.6.2.	Entorno <code>smallmatrix</code> (Matrices en línea)	65
4.7.	Creación de símbolos con comandos	66

1. Introducción

1.1. Creando un archivo

Lo primero que debemos hacer al iniciar un documento son escribir estas cuatro líneas básicas y completamente obligatorias a la hora de realizar un documento:

```
\documentclass[config]{tipo}

\begin{document}
    Texto
\end{document}
```

Un archivo en Latex se compone de dos partes. Un preámbulo y el cuerpo. El preámbulo es lo que va antes del `\begin{document}`, y el cuerpo es lo que va después.

1.2. Preámbulo

En el preámbulo se ponen todas las configuraciones del documento y los paquetes que se van a cargar para el mismo. Los paquetes son extensiones a la configuración que vienen por defecto. La forma de agregar un nuevo paquete es poner después de `documentclass` y antes del cuerpo del documento el comando `\usepackage{paquete}`, con el respectivo paquete que deseemos. A lo largo del documento se van a presentar los paquetes más comunes con sus funciones y configuraciones básicas.

1.2.1. Tipo de documento

Como vemos en las líneas que se pusieron anteriormente, lo esencial en el preámbulo es el **tipo**. El **tipo** que aparece hace referencia al tipo de documento que se está desarrollando. Dependiendo del tipo de documento que se escoja, se van a establecer ciertas configuraciones. Entre las más famosas tenemos:

tipo	Descripción
article	Para artículos académicos y otros documentos cortos que no necesitan dividirse en capítulos, sino que bastan las secciones y subsecciones y sus párrafos y subpárrafos
book	Para libros y otros documentos más largos que deben incluir capítulos, prólogo, apéndices o incluso partes
report	Para informes técnicos. Es similar a la clase <code>book</code>
memoir	Una clase todoterreno con un buen número de funciones adicionales integradas
beamer	Una clase para hacer diapositivas

La principal diferencia entre `book` y `report` es que la clase `Book` hace que todos los capítulos empiecen por una hoja impar, por lo que si un capítulo termina en una hoja impar, se va a producir un salto de una página para que el siguiente empiece en página impar. Esto no sucede en la clase `report`. Es por este tipo de diferencias que hay que escoger el tipo exacto para el trabajo que uno esté realizando

1.2.2. Configuración del documento

Recordando la línea que debemos poner en el preámbulo `\documentclass[config]{tipo}`, la otra opción configurable es `config`. Esta tiene diferentes opciones, por lo que se pueden poner tantas como sean necesarias, o ninguna, si es el caso:

config	Descripción
letterpaper, a4paper, ...	Aquí se selecciona el tamaño del papel. Se mostrarán los tamaños más adelante. El tamaño predefinido es letterpaper
landscape	Pone el documento de forma horizontal
10pt, 11pt, 12pt	El tamaño de la fuente. Puede ser 10pt, 11pt o 12pt
oneside, twoside	Indican si el documento debe estar adaptado a impresión por un sólo lado de la página o por ambos lados de ella
openright, openany	<code>openright</code> indica que los capítulos deben iniciar en páginas impares, mientras que <code>openany</code> indica que los capítulos pueden iniciar en cualquier página
onecolumn, twocolumn	Define si el documento va a estar escrito en una o dos columnas
fleqn	Esta opción hace que las ecuaciones se alineen a la izquierda, en vez de al centro, que es como se hace predeterminadamente

leqno	Con esta opción hacemos que el número de las ecuaciones quede alineado por la izquierda, en vez de al centro, que es como se hace predeterminadamente
draft, final	La opción draft se usa si queremos que la compilación del documento se haga a modo de “borrador”. Con draft haremos que las líneas que sean demasiado largas queden marcadas mediante cajas negras. La opción final producirá simplemente que el documento se compile de manera normal

Tipos de papel	
a4paper	Tamaño a4
letterpaper	Tamaño carta. 14 in x 8.5 in
a5paper	210 mm x 148 mm
b5paper	250 mm x 176 mm
executivepaper	10.5 in x 7.25 in

Las configuraciones predefinidas para la clase **book** son: **etterpaper**, 10pt, **twoside**, **onecolumn**, **final**, **openright**. Las configuraciones predefinidas para la clase **article** son: **letterpaper**, 10pt, **oneside**, **onecolumn**, **final**. Las configuraciones predefinidas para la clase **report** son: **letterpaper**, 10pt, **oneside**, **final**, **openany**.

1.3. Escritura en español (asentos y virgulilla)

Si queremos escribir un acento en Latex nos va a dar un error, ya que este lenguaje viene preconfigurado en el idioma inglés, por lo que no va a reconocer los caracteres especiales, como las letras con tildes o la ñe. Para solucionar esto, Latex nos permite indicarle manualmente cuando queremos que una de las letras lleve tilde, como veremos a continuación:

Ejemplo 1

Ví a María corriendo con una caña en la mano.

```
V\'i a Mar\'ia corriendo con una ca\~na en la mano.
```

Lo que tenemos que hacer si queremos usar los acentos es poner `\'` antes de la letra que queramos que lleve tilde, y un `\~` antes de la letra que queramos que lleve virgulilla.

1.3.1. Paquete `inputenc`

Este paquete gestiona las tildes, lo que permite escribirlas directamente sin tener que hacer uso del `\'`. Para usar este paquete debemos poner en el preámbulo `\usepackage[config]{inputenc}`. En la sección `config` debemos poner el codificador de estrada que se quiera. El del idioma español es `latin1`, aunque se recomienda ampliamente el uso de `utf8`, que aparte de codificar los acentos españoles, permite el uso de acentos de otros idiomas.

```
\usepackage[utf8]{inputenc}
\usepackage[latin1]{inputenc}
```

1.3.2. Paquete `babel`

Latex se encuentra configurado en idioma inglés, por lo que vamos a tener problemas a la hora de usar capítulos, ya que van a salir nombrados como *Chapter* y no como *Capítulo*. Otro problema que vamos a tener a la hora de usar Latex en idioma inglés es a la hora de la separación de palabras al final de un capítulo, ya que las palabras se separan de diferentes maneras dependiendo del idioma.

Para solucionar estos dos problemas, se debe usar el paquete `babel`. Cuando usemos el comando `\usepackage[idioma]{babel}` en el preámbulo podemos seleccionar el idioma en el que se va a encontrar el texto. El idioma que se recomienda es `spanish`, ya que es el español, aunque se pueden poner otro de los 30 idiomas que soporta el paquete.

1.4. Espacios

1.4.1. Espacios horizontales

Cuando escribimos en Latex y dejamos un espacio en blanco, Latex entiende que hay un espacio. Pero si dejamos más de uno, Latex entiende como si se estuviera dejando solo un espacio:

Ejemplo 1

Texto Texto Texto Texto Texto.

```
Texto Texto Texto Texto Texto.
```

Ejemplo 2

Texto Texto Texto Texto Texto.

```
Texto      Texto      Texto      Texto      Texto.
```

Si queremos agregar un espacio horizontal en una línea, debemos hacer uso del comando `\hspace{espacio}`, configurando `espacio` para señalar el espacio que deseamos. Por ejemplo:

Ejemplo 3

Texto Texto Texto Texto Texto.

```
Texto\hspace{1cm}Texto\hspace{1.5cm}Texto\hspace{2cm}Texto\hspace{3cm}Texto.
```

Otra opción que tenemos es el comando `\hfil`. Este comando nos permite empujar el texto hasta el final del párrafo:

Ejemplo 4

Texto
Texto Texto Texto

Texto.
Texto Texto Texto.

```
Texto\hfill Texto.\ \ Texto Texto Texto \hfill Texto Texto Texto  
.
```

1.4.2. Saltos verticales entre párrafos

Cuando queremos hacer un salto vertical entre párrafos se debe usar `\`, si y solo si los párrafos a separar se encuentran en la misma línea. veamos:

Ejemplo 1

Parrafo
Parrafo
Parrafo
Parrafo.

```
Parrafo\\Parrafo\\Parrafo\\Parrafo.
```

Si queremos separar dos párrafos basta con presionar la tecla **ENTER** hasta que entre los dos párrafos a separar haya una línea en blanco. Si no dejamos esta línea en blanco, Latex entiende que seguimos estando en el mismo párrafo:

Ejemplo 1

Parrafo
Parrafo Parrafo
Parrafo

```
Parrafo  
  
Parrafo  
Parrafo  
  
Parrafo
```

Cuando queremos indicarle a Latex que hemos acabado un párrafo debemos usar `\par`. Suele ocurrir que para indicar un salto de párrafo se use el `\\` porque visualmente crear un salto de párrafo más grande. Esto es incorrecto, porque deforma las cajas y puede ocasionar errores al momento de compilar. En el caso en que se quiera ver un espacio más grande se debe configurar en el preámbulo se debe usar el salto vertical después de terminar el párrafo con un `\par`.

1.4.3. Espacio entre párrafos con `setlength`

Para configurar el espaciado entre párrafos en el preámbulo se debe usar el comando `\setlength{\parskip}{tan}`. Por defecto el espacio entre párrafos es de una línea, por lo que *parskip* vale 0. Podemos cambiar su tamaño con el comando `\setlength`. Este comando debe ser escrito en el preámbulo. Si se hace

así va a afectar a todo el documento. La distancia que configuremos en **tamaño** se va a sumar a la medida de una línea, es decir, si escribimos en **tamaño 1cm**, el espacio entre párrafos va a ser de una línea más 1cm. El tamaño que las personas prefieren es de dos líneas en blanco, por lo que en la configuración del **tamaño** puede ponerse `\baselineskip`, que es el tamaño estándar de una línea.

Si no queremos que esta configuración afecte a todo el documento, sino solo a una porción de este, podemos poner el comando entre un entorno, o entre llaves. Veamos ejemplo del uso de este comando:

Ejemplo 1

Parrafo
Parrafo
Parrafo

```
\setlength{\parskip}{-2mm}  
Parrafo\par  
Parrafo\par  
Parrafo\par
```

Ejemplo 2

Parrafo

Parrafo

Parrafo

```
\setlength{\parskip}{5mm}  
Parrafo\par  
Parrafo\par  
Parrafo\par
```

1.4.4. Medida de saltos verticales

Ahora bien, se puede presentar el caso en el que queramos insertar un salto vertical con la medida que queramos. Para hacer esto debemos usar el comando `\vspace{espacio}`, configurando **espacio** para señalar el espacio que deseamos. Por ejemplo:

Ejemplo 1

Texto

Texto

Texto

```
Texto
\vspace{1in}

Texto
\vspace{1cm}

Texto
```

Si se quiere se pueden usar uno de los tres espacios que vienen configurados por defecto en Latex, que son:

<code>\smallskip</code>	<code>\medskip</code>	<code>\bigskip</code>
Texto	Texto	Texto
Texto	Texto	Texto

1.5. Alineación del texto

1.5.1. Alineación con comando

Latex justifica automáticamente los textos. Si queremos alinearlos a nuestro gusto tenemos dos opciones. La primera es declarando la variable `\centering`, `\raggedright` y `\raggedleft` para alinear el texto al centro, izquierda y derecha, respectivamente. El problema de usar estos comandos es que el texto desde el comando en adelante va a mantener esta alineación hasta el final del texto o hasta que haya otro comando que modifique lo mismo:

Ejemplo 1

Texto alineado a la izquierda

Texto centrado

Texto alineado a la derecha

```
\centering
Texto centrado

\raggedright
Texto alineado a la izquierda

\raggedleft
Texto alineado a la derecha
```

1.5.2. Alineación con entornos

La otra opción que tenemos para alinear un texto es crear un entorno. Los entornos sirven para que entre ellos funcione una configuración distinta a la del texto que está afuera. Por ejemplo, podemos crear un entorno donde lo que está dentro de él esté alineado a la derecha, mientras que el documento está justificado. Los entornos tienen la siguiente estructura:

```
\begin{tipo}
    Contenido del entrono
\end{tipo}
```

Los entornos que nos competen en esta sección son tres: `center`, `flushright` y `flushleft`. Cuando ponemos cualquiera de estas configuraciones dentro del `tipo` del entorno, lo que está adentro se alinea al centro, a la derecha y a la izquierda respectivamente:

Ejemplo 1

Texto centrado

Texto alineado a la derecha

Texto alineado a la izquierda

```
\begin{center}
    Texto centrado
\end{center}

\begin{flushright}
    Texto alineado a la derecha
\end{flushright}

\begin{flushleft}
    Texto alineado a la izquierda
\end{flushleft}
```

Como vemos, haciendo esto podemos hacer que solo una parte del texto a nuestra elección esté alineada, mientras que el resto del texto sigue estando justificado.

1.6. Letra

1.6.1. Tamaño de letra

El tamaño de la letra se puede configurar anteponiendo a un texto el comando `\tamaño` con el tamaño que se quiera. Si solo se quiere cambiar una palabra se puede encerrar entre corchetes `{\tamaño Texto}`:

\tamaño	Ejemplo	Texto
\tiny	tiny	Texto de ejemplo
\scriptsize	scriptsize	Texto de ejemplo
\footnotesize	footnotesize	Texto de ejemplo
\small	small	Texto de ejemplo
\normalsize	normalsize	Texto de ejemplo
\large	large	Texto de ejemplo
\Large	Large	Texto de ejemplo
\LARGE	LARGE	Texto de ejemplo
\huge	huge	Texto de ejemplo
\Huge	Huge	Texto de ejemplo

1.6.2. Estilo de letra

El estilo de la letra se puede configurar con el comando `\estilo{Texto}` con el estilo que se quiera. Por otra parte, para cambiar todo el texto se debe usar el siguiente comando `\estiloseries` o `\estilosshape`:

\estilosshape	\estilo	Ejemplo	Texto
\bfseries	\textbf	Negrita	Texto de ejemplo
\mdseries	\textmd	Negrita medio	Texto de ejemplo
\itshape	\textit	<i>Cursiva</i>	<i>Texto de ejemplo</i>
\slshape	\textsl	<i>Roman inclinado</i>	<i>Texto de ejemplo</i>
\scshape	\textsc	MAYÚSCULAS PEQUEÑAS	TEXTO DE EJEMPLO
\upshape	\textup	Rectas	Texto de ejemplo
	\underline	<u>Subrayado</u>	<u>Texto de ejemplo</u>
	\uppercase	MAYÚSCULAS	TEXTO DE EJEMPLO

1.6.3. Tipo de letra

De igual forma se puede cambiar el tipo de letra entre las siguientes con el comando `\tipo{Texto}`. Por otra parte, para cambiar todo el texto se debe usar el siguiente comando antes del texto `\tipofamily`:

<code>\tipofamily</code>	<code>\tipo</code>	Ejemplo	Texto
<code>\rmfamily</code>	<code>\textrm</code>	Roman	Texto de ejemplo
<code>\sffamily</code>	<code>\textsf</code>	Sans serif	Texto de ejemplo
<code>\ttfamily</code>	<code>\texttt</code>	Mecanografiado	Texto de ejemplo

Cabe recalcar que se pueden convidar tanto los tipos como los estilos de las letras, por ejemplo:

<code>\tipo</code>	Ejemplo
<code>\textrm{\Large \textbf{}}</code>	Roman Large Negrita
<code>\textsf{\scriptsize \underline{}}</code>	<u>Sans serif Scriptsize Subrayado</u>

1.6.4. Interlineado

Para modificar el interlineado se debe usar el paquete `setspace`. Este paquete nos permite escoger el interlineado entre sus tres opciones predefinidas o entre una a nuestra elección. Para usarlo debemos definir el paquete `\usepackage{setspace}` en el preámbulo. Cuando hayamos hecho eso se tienen los siguientes comandos, que pueden ser puestos antes de un párrafo para que afecte desde ese punto en adelante, o entre un entorno para que solo modifique el interior de este:

Tamaño	Ejemplo
<code>\doublespacing</code>	Interlineado de 2
<code>\onehalfspacing</code>	Interlineado de 1.5
<code>\singlespacing</code>	Interlineado de 1

2. Haciendo un taller

2.1. Portada

2.1.1. Portada con `\maketitle`

Para hacer una portada con el comando `\maketitle` debemos definir en el preámbulo tres parámetros: `\title{titulo}`, donde definimos el título del documento, `\author{autor}` donde definimos el nombre del autor y `\date{fecha}`, donde definimos la fecha de creación del documento. Una vez tenemos definidos estos parámetros, cuando estemos dentro del cuerpo del documento debemos escribir `\maketitle`, lo que automáticamente creará una portada. Dependiendo del tipo de documento la portada puede cambiar. Por ejemplo, en la clase `book` o `report` el título aparecerá al principio y en una página aparte. En cambio, con la clase `article` el título aparecerá en la parte superior de la primera página del documento. Si queremos que en la clase `article` aparezca el título en una página aparte, debemos especificar la opción `titlepage`, que está desactivada por defecto. El tamaño de letra del título es `\LARGE`, pero puede ser cambiada dentro de `\title{titulo}`. Veamos unos ejemplos:

Ejemplo 1

Titulo del documento

Autor del documento

8 de noviembre de 2021

```
\documentclass{article}

\title{\Huge Titulo del documento}
```



```
\author{Autor del documento}
\date{\today}

\begin{document}
    \maketitle
\end{document}
```

Como vemos en el ejemplo, como fecha se puso `\today`. Este comando define la fecha del día en que se compila el documento.

2.1.2. Portada con entorno *titlepage*

Si no nos gusta la portada que nos genera Latex podemos hacer una a nuestro gusto. Esto se puede lograr con el entorno *titlepage*. Dentro de este entorno podemos escribir como queremos que puede ser la portada. Veamos un ejemplo:

Ejemplo 1

Universidad

PROYECTO FINAL

HECHO POR:

Autor

8 de noviembre de 2021

```
\documentclass{article}

\begin{document}
  \begin{titlepage}
    \centering
    \vspace{50pt}
    {\scshape\Huge \textbf{Universidad}}
    \vspace{100pt}

    {\scshape\huge Proyecto final}
    \vspace{80pt}

    {\scshape\Large Hecho por:}
```

```

        \vspace{10pt}

        {\large Autor}
        \vspace{22pt}

        {\large \today}
        \vspace{50pt}
    \end{titlepage}
\end{document}

```

2.2. Listas

2.2.1. Entorno *itemize*

Para crear una lista de elementos se debe usar el entorno `\begin{itemize}`. El esquema de este entorno es el siguiente:

```

\begin{itemize}
    \item Texto
    \item ...
    ...
\end{itemize}

```

Enfrente de cada `\item` escribimos el texto que va a estar listado. Veamos un ejemplo:

Ejemplo 1

- Item 1
- Item 2
- Item 3

De igual manera podemos crear un ítem dentro de los ítems:

Ejemplo 2

- Item 1
- Item 2
 - Item 2.1
 - Item 2.1
- Item 3

```
\begin{itemize}
  \item Item 1
  \item Item 2
    \begin{itemize}
      \item Item 2.1
      \item Item 2.1
    \end{itemize}
  \item Item 3
\end{itemize}
```

2.2.2. Cambio de enumeración individual *itemize*

Si no nos gustan los cuadrados que viene por defecto podemos cambiar individualmente la notación de cada ítem si ponemos después del `\item` unos corchetes cuadrados indicando lo que queremos que se muestre:

Ejemplo 1

- 1. Item 1
- A- Item 2
- i) Item 3

```
\begin{itemize}
  \item[1.] Item 1
  \item[A-] Item 2
  \item[i)] Item 3
\end{itemize}
```

2.2.3. Entorno *enumerate*

El entorno `enumerate` funciona igual que el `itemize`, con la única diferencia de que este entorno enumera los ítems, y no los lista. Esto quiere decir que hace una lista numérica con todos los ítems. Su formato es igual al del entorno `itemize`:

Ejemplo 1

1. Item 1
2. Item 2
3. Item 3

```
\begin{enumerate}  
  \item Item 1  
  \item Item 2  
  \item Item 3  
\end{enumerate}
```

También se pueden crear `enumerate` dentro de otros, y combinarlas con `itemize`. Veamos:

Ejemplo 2

1. Item 1
2. Item 2
 - Item 2.1
 - Item 2.2
 - Item 2.2.1
 - Item 2.2.2
 - Item 2.3
 - Item 2.4
 - a) Item 2.4.1
 - b) Item 2.4.2

3. Item 3

4. Item 4

```
\begin{enumerate}
  \item Item 1
  \item Item 2
    \begin{itemize}
      \item Item 2.1
      \item Item 2.2
        \begin{enumerate}
          \item Item 2.2.1
          \item Item 2.2.2
        \end{enumerate}
      \item Item 2.3
      \item Item 2.4
        \begin{enumerate}
          \item Item 2.4.1
          \item Item 2.4.2
        \end{enumerate}
    \end{itemize}
  \item Item 3
  \item Item 4
\end{enumerate}
```

2.2.4. Cambio de enumeración individual *enumerate*

Al igual que con el entorno `itemize` podemos cambiar individualmente la enumeración de la lista.

Ejemplo 1

1. Item 1

2. Item 2

I) Item 3

a: Item 4

3. Item 5

4. Item 6

```
\begin{enumerate}
  \item Item 1
  \item Item 2
  \item[I)] Item 3
  \item[a:] Item 4
  \item Item 5
  \item Item 6
\end{enumerate}
```

2.2.5. Cambio de enumeración global

Si queremos modificar completamente la forma como se numera debemos usar el paquete `\usepackage{enumerate}`. Este paquete nos permite seleccionar que tipo de enumeración queremos para el entorno `enumerate`. Para modificar esto tenemos escoger el tipo de enumeración después de crear el entorno: `\begin{enumerate}[enumeration]`. En la casilla `enumeration` escribimos como queremos que sea la secuencia de la enumeración. Veamos unos ejemplos:

Ejemplo 1

a: Item 1

b: Item 2

1- Item 1

2- Item 2

3- Item 3

c: Item 3

d: Item 4

```
\begin{enumerate}[a:]
  \item Item 1
  \item Item 2
\end{enumerate}
```

```

\begin{enumerate}[1-]
  \item Item 1
  \item Item 2
  \item Item 3
\end{enumerate}
\item Item 3
\item Item 4
\end{enumerate}

```

Ejemplo 2

I) Item 1

II) Item 2

$\Delta+$ Item 1

$\Delta+$ Item 2

$\Delta+$ Item 3

III) Item 3

IV) Item 4

```

\begin{enumerate}[I)]
  \item Item 1
  \item Item 2
    \begin{enumerate}[$\Delta$+]
      \item Item 1
      \item Item 2
      \item Item 3
    \end{enumerate}
  \item Item 3
  \item Item 4
\end{enumerate}

```


2.3. Citación y bibliografía

Para citar y referenciar vamos a usar dos paquetes. El primero, `apacite` que nos permite hacer referencias en apa, y `natbib`, que nos amplía las formas de citar en apa. Aparte de esto se va a usar un archivo `bib.bib` donde se almacenarán las citas.

2.3.1. Archivo `bib.bib`

Para almacenar las citas vamos a crear un archivo llamado `bib` (u otro nombre) con la extensión `.bib`. En este archivo se van a almacenar las citas, que posteriormente se vayan utilizando en el texto. Las citas que estén dentro de este documento deben cumplir un formato, y deben llevar ciertos campos dependiendo de su tipo. Se recomienda que este archivo se guarde en la raíz de la carpeta del trabajo, aunque no es completamente necesario. Veamos un ejemplo del archivo:

Si queremos citar un capítulo de una revista, los campos imprescindibles son: nombre del autor, año de publicación, nombre del artículo, nombre de la revista, volumen y número de la revista y las páginas que componen el artículo. Cuando tengamos esa información la debemos adjuntar en el archivo `bib.bib` con el siguiente formato:

```
@article{ref,
  author = {Apellido, Nombre Nombre},
  journal = {Nombre de la revista},
  number = {Num},
  pages = {pag},
  title = {Titulo del articulo},
  volume = {vol},
  year = {2020}
}
```

El campo `ref` se refiere a un nombre que le asignamos a la cita para llamarla cuando lo necesitamos en el cuerpo del archivo. Es importante asignarle a cada cita un nombre diferente.

Como vemos en el ejemplo, esta es la organización básica de un artículo de revista. El nombre que aparece después del `@` es el tipo de cita. Dependiendo del tipo, debemos poner más o menos campos. A continuación, se presentará una tabla con los tipos más comunes de citas y los campos que se sugieren deben contener:

Se marcarán con una x los campos que deben incluir

Tipo Campos	article	magazine	newspaper	book	misc	misc (pag. web)
address				x	x	
author	x	x	x	x	x	x
doi	x	x	x	x	x	
edition				x	x	
editor	x	x	x	x	x	
journal	x	x	x			
number	x	x	x	x	x	
pages	x	x	x	x	x	
publisher				x	x	
title	x	x	x	x	x	x
translator	x	x	x	x		
url	x	x	x	x	x	x
urldate	x	x	x	x	x	x
volume	x	x	x	x	x	
year	x	x	x	x	x	x

La categoría `misc` se usa para todo lo que no cabe en alguna categoría. En caso de que se quiera citar una página web se debe usar la categoría `misc` y se deben llenar los campos que se sugirieron en el cuadro como `misc (pag. web)`.

Es importante aclarar que no es necesario que se llenen todos campos de para que se haga una cita, pero entre más datos se tenga, mejor será la cita. Si no se completa alguno de los casos Latex automáticamente completará el campo con un `s.f.` en el caso de las fechas, y su equivalente a cualquiera de los otros campos.

Ejemplo 1

Gomez, J. (2020). Titulo. Descargado 25/02020, de www.titulo.com

```
@misc{app,
author = {Gomez, Juan},
title = {Titulo},
url = {www.titulo.com},
urldate = {25/02020},
```

```
year = {2020}  
}
```

Ejemplo 2

Gomez, J. (s.f.). Titulo. Descargado 25/02020, de www.titulo.com

```
@misc{app,  
author = {Gomez, Juan},  
title = {Titulo},  
url = {www.titulo.com},  
urldate = {25/02020},  
}
```

Ejemplo 3

Titulo.(s.f.). Descargado 25/02020, de www.titulo.com

```
@misc{app,  
title = {Titulo},  
url = {www.titulo.com},  
urldate = {25/02020},  
}
```

2.3.2. Referencias

Cuando tengamos todas nuestras citas en el archivo `bib.bib`, podremos empezar a citar en el cuerpo del documento. Para esto debemos escribir en el cuerpo del trabajo dos comandos: `\bibliographystyle{apacite}` y `\bibliography{bib.bib}`. El primero señalaremos el estilo de citación, que en este caso es `apa`. En el segundo comando señalamos dónde se encuentra el documento `bib.bib` o el archivo donde se encuentras las bibliografías. Si se escogió otro nombre para el documento bibliográfico diferente de `bib.bib`, se debe modificar el interior de los corchetes con el respectivo nombre o ruta del archivo de bibliografía, por ejemplo `\bibliography{ref.bib}`.

Cuando tengamos lo anterior listo, podemos empezar a citar. Para hacer esto tenemos que usar

uno de los comandos `\cite` que se van a presentar a continuación. Dependiendo del que se use se va a mostrar más o menos información.

Comando	Descripción	Ejemplo
<code>\citet {ref}</code>	Citación textual	Apellido (año) Apellido y cols., (año)
<code>\citep {ref}</code>	Citación con paréntesis	(Apellido, año) (Apellido y cols., año)
<code>\citealp {ref}</code>	Igual que <code>\citep</code> pero sin usar paréntesis	Apellido, año Apellido y cols., año
<code>\citet *{ref}</code>	Igual que <code>\citet</code> pero si son muchos autores, lo muestra todos	Apellido, Apellido, y Apellido (año)
<code>\citep *{ref}</code>	Igual que <code>\citep</code> pero si son muchos autores, lo muestra todos	(Apellido, Apellido, y Apellido, año)
<code>\citeauthor {ref}</code>	Solo cita el autor	Apellido y cols.
<code>\citeyear {ref}</code>	Solo cita el año	año
<code>\citeyearpar {ref}</code>	Solo cita el año con paréntesis	(año)

Para citar en el texto debemos usar una de las anteriores opciones, reemplazando el **ref** por el nombre que le asignamos a la referencia. Veamos un ejemplo teniendo en cuenta la siguiente referencia agregada al archivo `bib.bib`:

```
@article{Inhumanas ,
author = {Cruz Kronfly , Fernando},
journal = {Cuadernos de {A}dministracion},
number = {27},
pages = {14-22},
title = {El mundo del trabajo y las organizaciones desde la
perspectiva de las practicas inhumanas.},
volume = {18},
year = {2002}
}
```

Ejemplo 1

Por otro lado, es importante siempre tener esto en cuenta, ya que en las empresas debe existir en cierta medida un poco de deshumanización, ya que es necesario marcar la diferencia entre los altos y los bajos mandos (Krofly, 2002, p. 21). Esta diferenciación debe hacerse con humanidad ante

todo. Como menciona Kronfly, “existen dirigentes auténticos que cuya autoridad sobre los demás resulta inobjetable debido a la transparencia de sus fundamentos y la legitimidad aceptación por parte de los dirigidos” (2002, p. 21).

```
Por otro lado, es importante siempre tener esto en cuenta, ya
que en las empresas debe existir en cierta medida un poco de
deshumanizacion, ya que es necesario marcar la diferencia
entre los altos y los bajos mandos \citep[p. 21]{Inhumanas}.
Esta diferenciacion debe hacerse con humanidad ante todo.
Como menciona Kronfly, ‘‘existen dirigentes autenticos que
cuya autoridad sobre los demas resulta inobjetable debido a
la transparencia de sus fundamentos y la legitimidad
aceptacion por parte de los dirigidos’’ \citeyearpar[p. 21]{
Inhumanas}.
```

2.3.3. Bilbiografía

2.3.4. Sangría para citas (quote)

2.3.5. Paquete hyperref

Este paquete nos permite linkear las partes de nuestro documento

Con `\url{URL}` insertamos el URL, y con `\href{URL}{text}` insertamos un URL con un nombre distinto

2.4. Identación o sangría

2.4.1. Identación para párrafos

2.5. Formato secciones, subsecciones y subsubsecciones

Para dar formato a los párrafos se debe usar el paquete `\usepackage{titlesec}`. Este permite establecer el formato de las secciones, subsecciones y subsubsecciones. En el preámbulo se pueden

escribir dos funciones: `\titleformat*` o `\titleformat` para modificar los formatos.

2.5.1. `titleformat*`

Empecemos con `\titleformat*{comando}{formato}`. Este comando solo cambia lo necesario. Este tiene dos parámetros: `{comando}` y `{formato}`. `{comando}` se refiere al comando que se va a modificar. Puede ser `\section`, `\subsection` o `\subsubsection`. Finalmente `{formato}` se refiere al formato que se le va a dar. Pueden cambiarse los tipos de letra, colores, entre otros:

Ejemplo 1

1. Titulo

```
\titleformat*{\section}{\large \bfseries}
\section{Titulo}
```

Ejemplo 2

1. Titulo

```
\titleformat*{\section}{\LARGE \ttfamily \centering \color{red}}
\section{Titulo}
```

2.5.2. `titleformat`

Este formato funciona igual que el anterior, pero tiene más opciones: `\titleformat{comando}[forma]{formato}{label}{paso}{despues}{antes}`. En la opción `{comando}` se escribe el comando a modificar, que puede ser `\section`, `\subsection` o `\subsubsection`. En la opción `[forma]` se selecciona el formato de la sección. Se recomienda poner siempre `block`. La opción `{formato}` se usa para establecer el formato como en el punto anterior. En la opción `{label}` se pone la enumeración que se quiere para las secciones. Puede ser en estilo con números romanos usando `\Roman{\section}` o normal usando `\arabic{\section}` para enumerar con número. La opción

{paso} es la separación entre el label y el texto. La opción {antes} permite poner comando adicionales, pero se recomienda dejarla en blanco.

Ejemplo 1

I. Titulo

```
\titleformat{\section}[block]{\large \bfseries \color{blue}}{\Roman{section}.}{2cm}{}
\section{Titulo}
```

Ejemplo 2

1. Titulo

```
\titleformat{\section}[block]{\LARGE \bfseries}{\arabic{section}.}{1cm}{}
\section{Titulo}
```

2.5.3. titlespacing

Este formato se usa para modificar los espacios de las secciones, subsecciones y subsubsecciones. Tenemos a `\titlespacing{comando}{izquierda}{antes}{despues}`. La opción {comando} se usa para poner el comando a modificar, como `\section`, `\subsection` o `\subsubsection`. La opción {izquierda} modifica el margen izquierdo. La opción `antes` modifica el espacio vertical del texto antes de la sección y la sección y la opción {despues} modifica el espacio entre la sección y el texto a continuación.

Para los ejemplos se va a utilizar el paquete `\usepackage{lipsum}`, que permite poner párrafos de texto de ejemplo.

Ejemplo 1

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer

tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

1. Titulo

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\titleformat*{\section}{\large \bfseries}
\titlespacing{\section}{0cm}{0cm}{0cm}

\lipsum[4]

\section{Titulo}

\lipsum[2]
```

Ejemplo 2

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

1. Titulo

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\titleformat*{\section}{\large \bfseries}
\titlespacing{\section}{3cm}{2cm}{1cm}

\lipsum[4]

\section{Titulo}

\lipsum[2]
```

3. Tablas

3.1. Creación de tablas

El formato básico para la creación de una tabla usando el entorno `tabular` es:

```
\begin{tabular}{columnas}  
  Columna 1 & Columna 2 & etc...\n  
  ...  
  Columna 1 & Columna 2 & etc...\n  
  \hline  
  Columna 1 & Columna 2 & etc...\n  
  ...  
\end{tabular}
```

Las columnas se escriben indicando la alineación de estas. Por ejemplo, si se quieren dos columnas alineadas a la izquierda, se escribe `{l l}`, y si se quieren tres columnas, donde la primera se alinee al centro y las otras dos a la derecha se escribe `{c r r}`. Si se quieren separar las columnas por líneas verticales se separan las columnas por un signo `|`, por ejemplo, si se quieren separar dos columnas por una línea se escribe `{c | c}`, y si también se quieren crear las líneas exteriores de estas mismas dos columnas se escribe `{| c | c |}`. Finalmente, para hacer líneas verticales escribimos entre dos columnas el comando `\hline`

Ejemplo 1

Columna 1	Columna 2	Columna 3
TextoTextoTexto	TextoTextoTexto	TextoTextoTexto

```
\begin{tabular}{|l|c|r|}  
  \hline  
  Columna 1 & Columna 2 & Columna 3\\  
  \hline  
  TextoTextoTexto & TextoTextoTexto & TextoTextoTexto\\  
  \hline  
\end{tabular}
```

Se pueden poner doble y triple las líneas que separan tanto a las columnas como a las filas:

Ejemplo 2

Columna 1	Columna 2	Columna 3
TextoTextoTexto	TextoTextoTexto	TextoTextoTexto

```
\begin{tabular}{r||r|||r}
  Columna 1 & Columna 2 & Columna 3\\
  \hline
  TextoTextoTexto & TextoTextoTexto & TextoTextoTexto\\
\end{tabular}
```

Ejemplo 3

Columna 1	Columna 2
TextoTextoTexto	TextoTextoTexto
TextoTextoTexto	TextoTextoTexto
TextoTextoTexto	TextoTextoTexto

```
\begin{tabular}{||c||c||}
  \hline
  Columna 1 & Columna 2\\
  \hline
  \hline
  TextoTextoTexto & TextoTextoTexto\\
  \hline
  TextoTextoTexto & TextoTextoTexto\\
  \hline
  TextoTextoTexto & TextoTextoTexto\\
  \hline
\end{tabular}
```

3.2. Tamaño de columnas

Cabe recalcar que en las tablas se puede establecer el tamaño de las columnas. En la configuración `columnas` se debe agregar `ptamaño`, con el tamaño que se quiera.

Ejemplo 1

Columna 1	Columna 2
-----------	-----------

```
\begin{tabular}{|p{80pt}|p{200pt}|}  
  \hline  
  Columna 1 & Columna 2\\  
  \hline  
\end{tabular}
```

Ejemplo 2

Columna 1	Columna 2
-----------	-----------

```
\begin{tabular}{|p{300pt}|p{150pt}|}  
  \hline  
  Columna 1 & Columna 2\\  
  \hline  
\end{tabular}
```

3.3. Agrupación de filas

Para realizar esto se necesita el paquete `\usepackage{multirow}`. Para agrupar una fila se emplea el comando `\multirow{filas}{tamaño}{texto}`. En el apartado `fila` se seleccionan cuantas filas se quieren agrupar. En `tamaño` se escoge el tamaño que va a tener la fila agrupada. Se sugiere poner un asterisco `*` para que ajuste el tamaño de acuerdo con las otras tablas. Finalmente, en el apartado `texto` pone el texto que va a contener la fila agrupada. Veamos ejemplos:

Ejemplo 1

Columna 1	Columna 2
	Texto
Texto	Texto
	Texto

```
\begin{tabular}{|c|c|}
  \hline
  Columna 1 & Columna 2\\
  \hline
  \multirow{2}{*}{Texto} & Texto\\
  \hline
  & Texto\\
  & Texto\\
  \hline
\end{tabular}
```

Como vemos, no podemos hacer un `\hline` porque es una línea horizontal tal que va desde el inicio hasta el final de la tabla. Para solucionar esto podemos emplear el `\cline{columna1-columna2}`, donde seleccionamos una columna inicial `columna1` y una columna final `columna2` para trazar una línea. La línea se traza desde el inicio de la columna inicial `columna1` hasta el final de la columna final `columna2`. Veamos:

Ejemplo 2

Columna 1	Columna 2
Texto	Texto
	Texto
	Texto

```
\begin{tabular}{|c|c|}
  \hline
  Columna 1 & Columna 2\\
  \hline
  \multirow{3}{*}{Texto} & Texto\\
  \cline{2-2}
  & Texto\\
  & Texto\\
\end{tabular}
```

```

\cline{2-2}
& Texto\\
\hline
\end{tabular}

```

Como vemos en el ejemplo, se hicieron dos líneas, desde el inicio de la columna 2 hasta el final de la columna 2. Veamos otro ejemplo:

Ejemplo 3

Columna 1	Columna 2	Columna 3

```

\begin{tabular}{|c|c|c|}
\hline
Columna 1 & Columna 2 & Columna 3\\
\hline
\multirow{5}{*}{} & \multirow{5}{*}{} & \multirow{5}{*}{}\\
\cline{1-1} \cline{3-3}
& & \\
\cline{1-2}
& & \\
\cline{2-3}
& & \\
\cline{3-3}
& & \\
\hline
\end{tabular}

```

3.4. Agrupación de columnas

Para agrupar una columna se emplea el comando `\multicolumn{columnas}{posición}{texto}`. En el apartado `columnas` se seleccionan cuántas columnas se desean agrupar. En `posición` se escoge la posición del texto en la columna, y en `texto` se inserta el texto que irá dentro de la columna. Veamos ejemplos:

Ejemplo 1

Columna 1	Columna 2
Texto	Texto
Texto	
Texto	Texto

```
\begin{tabular}{|c|c|}  
  \hline  
  Columna 1 & Columna 2\\  
  \hline  
  Texto & Texto\\  
  \hline  
  \multicolumn{2}{|c|}{Texto}\\  
  \hline  
  Texto & Texto\\  
  \hline  
\end{tabular}
```

Ejemplo 2

Columna 1	Columna 2	Columna 3
Texto	Texto	
Texto		
Texto		
Texto		
Texto		Texto

```

\begin{tabular}{|c|c|c|}
\hline
Columna 1 & Columna 2 & Columna 3\\
\hline
Texto & \multicolumn{2}{c|}{Texto}\\
\hline
\multicolumn{3}{|l|}{Texto}\\
\hline
\multicolumn{3}{|c|}{Texto}\\
\hline
\multicolumn{3}{|r|}{Texto}\\
\hline
\multicolumn{2}{|c|}{Texto} & Texto\\
\hline
\end{tabular}

```

3.5. Color en las tablas

Para colorear las tablas debemos usar los paquetes `\usepackage{colortbl}` y `\usepackage{array}`. Aquí también se pueden usar los colores personalizados.

3.5.1. Color en las columnas

Para colorear las columnas debemos usar el comando `\columncolor[modelocolor]{color}`, donde `modelocolor` corresponde al modelo del color que se va a usar, puede ser `rgb`, `cmyk` o `gray`. El apartado `color` especifica el color en el respectivo modelo. Hay colores ya definidos, que son: `black`, `white`, `red`, `green`, `blue`, `cyan`, `magenta` y `yellow`. Para definir el color debemos hacer uso de las propiedades del paquete `array`, poniendo el comando en la configuración de la tabla. Por ejemplo:

Ejemplo 1

Columna 1	Columna 2	Columna 3	Columna 4
Texto	Texto	Texto	Texto


```

\begin{tabular}{|>\columncolor[rgb]{0.8,0,0.2}}c|>\columncolor[cm
y]{0.8,0.4,0.4,0.1}}c|>\columncolor[gray]{0.8}}c|>\columncolor{green}}c|}
\hline
Columna 1 & Columna 2 & Columna 3 & Columna 4\\
\hline
Texto & Texto & Texto & Texto\\
\hline
\end{tabular}

```

3.5.2. Color en las filas

Para colorear las filas se debe usar el comando `\rowcolor[modelocolor]{color}`. Los ajustes de `modelocolor` y `color` son los mismos que el punto anterior, el color en las columnas. Por ejemplo:

Ejemplo 1

Columna 1	Columna 2
Texto	Texto

```

\begin{tabular}{|c|c|}
\hline
\rowcolor[rgb]{0.3,0.6,0.1} Columna 1 & Columna 2\\
\hline
\rowcolor{yellow} Texto & Texto\\
\hline
\end{tabular}

```

3.5.3. Color en las celdas individuales

Para colorear una celda debemos usar el comando `\cellcolor[modelocolor]{color}`. El `modelocolor` y el `color` se configurar igual que en los dos puntos anteriores. Veamos un ejemplo:

Ejemplo 1

Columna 1	Columna 2
Texto	Texto
Texto	Texto
Texto	Texto

```
\begin{tabular}{|c|c|}
\hline
\rowcolor[rgb]{0.3,0.6,0.1} Columna 1 & Columna 2\\
\hline
\rowcolor{yellow} Texto & Texto\\
\hline
\end{tabular}
```

3.6. El paquete *array*

Para usarlo ponemos en el preámbulo `\usepackage{array}`. Este paquete permite aumentar las opciones dentro del entorno tabular. Tiene dos usos importantes, el primero es establecer columnas del tamaño deseado, y el segundo, insertar declaraciones antes de la configuración de una columna.

3.6.1. Definir tamaño de las columnas

En el primer caso, cuando se quiere insertar una columna del tamaño deseado se debe poner en la configuración de la columna `m{medida}`, estableciendo la medida deseada. La columna se alinea a la izquierda:

Ejemplo 1

Columna 1	Columna 2
Texto Texto Texto Texto Tex- to Texto	Texto Tex- to Texto Texto Tex- to Texto

```
\begin{tabular}{|m{5cm}|m{2cm}|}
```

```

\hline
Columna 1 & Columna 2\\
\hline
Texto Texto Texto Texto Texto Texto & Texto Texto Texto
    Texto Texto Texto\\
\hline
\end{tabular}

```

Ejemplo 2

Columna 1	Columna 2
Texto Texto Tex- to Texto Texto Texto	Texto Texto Texto Texto Texto Texto

```

\begin{tabular}{|m{3cm}|m{8cm}|}
\hline
Columna 1 & Columna 2\\
\hline
Texto Texto Texto Texto Texto Texto & Texto Texto Texto
    Texto Texto Texto\\
\hline
\end{tabular}

```

3.6.2. Declaraciones en las columnas

Este paquete también nos permite agregar declaraciones en la configuración de las columnas. Esto se hace agregando `>\{declaración\}` o `<\{declaración\}` antes de la configuración de la columna para que inserte la declaración antes o después de la configuración respectivamente. Vamos:

Ejemplo 1

Columna 1	<i>Columna 2</i>
Texto Texto Texto Texto Texto Texto	<i>Texto Texto Texto Texto Texto Texto</i>

```

\begin{tabular}{|>\bfseries c|>\slshape c|}
\hline
Columna 1 & Columna 2\\
\hline
Texto Texto Texto Texto Texto Texto & Texto Texto Texto
      Texto Texto Texto\\
\hline
\end{tabular}

```

Por supuesto, las dos anteriores configuraciones se pueden combinar:

Ejemplo 2

COLUMNA 1	Columna 2
TEXT0 TEXT0 TEXT0 TEXT0 TEXT0 TEXT0	Text0 Text0 Text0 Text0 Text0 Text0

```

\begin{tabular}{|>\scshape m{8cm}|>\ttfamily \bfseries m{3cm}
|}
\hline
Columna 1 & Columna 2\\
\hline
Texto Texto Texto Texto Texto Texto & Texto Texto Texto
      Texto Texto Texto\\
\hline
\end{tabular}

```

3.7. Entorno *tabularx*

Tabularx es un paquete adicional que debe ser agregado al preámbulo del documento. Este paquete necesita del paquete `array` para su funcionamiento. Es decir, es necesario agregar al preámbulo los paquetes `tabularx` y `array`:

```
\usepackage{tabularx} y \usepackage{array}
```

3.7.1. Tamaño de la tabla

Este paquete agrega una nueva configuración para hacer tablas. Se trata de **X**. Cuando se usa en una tabla en el entorno **tabularx** se especifica que esa columna debe ajustar automáticamente su ancho para que la tabla ocupe la totalidad de del ancho de la tabla definida anteriormente. Su estructura es la siguiente:

```
\begin{tabularx}{ancho}{columnas}
...
\end{tabularx}
```

El ancho correspondo al ancho de la tabla, y las columnas se configuran como si se tratara del entorno **tabular**. De igual forma el contenido de este es similar al del entorno **tabular**. Veamos un ejemplo del tamaño. Recordemos que para que el tamaño se ajuste debemos establecer como alineación en la columna que queremos que se ajuste la letra **X**:

Ejemplo 1

Columna 1	Columna 2
Texto Texto Texto Texto Texto Texto	Texto Texto Texto Texto Texto Texto

```
\begin{tabularx}{300pt}{|X|c|}
\hline
Columna 1 & Columna 2\\
\hline
Texto Texto Texto Texto Texto Texto & Texto Texto Texto  

  Texto Texto Texto\\
\hline
\end{tabularx}
```

Ejemplo 2

Columna 1	Columna 2
Texto Texto Texto Texto Texto Texto	Texto Texto Texto Texto Texto Texto

```

\begin{tabularx}{300pt}{|c|X|}
\hline
Columna 1 & Columna 2\\
\hline
Texto Texto Texto Texto Texto Texto & Texto Texto Texto
      Texto Texto Texto\\
\hline
\end{tabularx}

```

Ejemplo 3

Columna 1	Columna 2
Texto Texto Texto Tex- to Texto Texto	Texto Texto Texto Tex- to Texto Texto

```

\begin{tabularx}{250pt}{|X|X|}
\hline
Columna 1 & Columna 2\\
\hline
Texto Texto Texto Texto Texto Texto & Texto Texto Texto
      Texto Texto Texto\\
\hline
\end{tabularx}

```

Como vemos en los ejemplos, las columnas que se establecen con el valor X se transforman para que la tabla mantenga el tamaño que se indicó. Para evitar transformaciones y que todas las columnas se mantengas del mismo tamaño, se debe establecer a X como la alineación de todas las columnas.

3.7.2. Tabla a tamaño completo

Se puede establecer como tamaño de la tabla el tamaño del texto, poniendo en el parametro ancho el comando `\textwidth`. Si se hace eso la tabla queda así:

Ejemplo 1

Colomna 1	Columna 2
Texto Texto Texto Texto Texto Texto	Texto Texto Texto Texto Texto Texto

```
\begin{tabularx}{\textwidth}{|X|X|}
\hline
Colomna 1 & Columna 2\\
\hline
Texto Texto Texto Texto Texto Texto & Texto Texto Texto
      Texto Texto Texto\\
\hline
\end{tabularx}
```

Por defecto la configuración `X` alinea el texto a la izquierda. Si se quiere cambiar la configuración, se puede hacer haciendo uso de la agrupación de columnas `\multicolumn`. Por ejemplo:

Ejemplo 2

	Columna 1	Columna 2
	Texto Texto	Texto Texto
Texto Texto		Texto Texto

```
\begin{tabularx}{300pt}{|X|r|}
\hline
\multicolumn{1}{|r|}{Columna 1} & Columna 2\\
\hline
\multicolumn{1}{|c|}{Texto Texto} & Texto Texto\\
\hline
Texto Texto & Texto Texto\\
\hline
\end{tabularx}
```

3.8. Posicionar tablas (Tablas flotantes)

Puede suceder que las tablas no queden exactamente dónde queremos. Para solucionar esto debemos meter la tabla dentro de un elemento flotante que nos deje posicionarlo con precisión:

```

\begin{table}[ubicacion]
  \begin{tabular}{columnas}
    ...
  \end{tabular}
\end{table}

```

Como vemos en el ejemplo, lo que se hizo fue crear el entorno `table` y dentro de ella se creó la tabla con el entorno `tabular`. El entorno `table` es un entorno flotante, por lo que podemos indicar en qué lugar queremos que se posicione. Para seleccionar la ubicación que queremos debemos indicarla en su configuración en `ubicacion`. Las ubicaciones que podemos escoger son las siguientes:

ubicacion	Posición
b	Al fondo de la página
h	En la misma posición donde se encuentra en el código
t	Al principio de la página
p	En una página que solo contenga elementos flotantes
!	Ignora la mayoría de las restricciones que pone Latex

Normalmente uno va a querer que la tabla se posicione donde uno la ha escrito en el código, por lo que se recomienda poner en la configuración `[!h]`, para indicar a Latex que la ponga justo en el lugar donde está en el código y que ignore la mayoría de restricciones.

Hay casos donde puede resultar imposible poner la tabla justo donde se quiere. Para evitar errores al momento de compilar el programa, Latex permite poner más de una posición, por lo que la configuración más recomendada es `[!hbt]`. Esto quiere decir que, si la tabla no se puede poner en el mismo lugar, entonces que se ponga al fondo de la página. Si esto no es posible, entonces se pondrá al principio de la página.

3.9. Entorno *longtable*

Cuando generamos una tabla de más de una página Latex suele cometer errores al hacerla. Es por esto por lo que se debe usar el paquete `\usepackage{longtable}`. Este paquete permite crear tablas que ocupan varias páginas.

El formato de la tabla es el siguiente:


```

\begin{longtable}[alineacion]{columnas}
  \hline
  Contenido de la tabla\\
  \hline
\end{longtable}

```

3.9.1. Tablas sin encabezados y pies en todas las páginas

Como podemos ver, el formato de esta tabla es como el formato de una tabla cualquiera, por lo que el parámetro `columnas` se configura el igual que se hace con el entorno `tabular`. La novedad que nos presenta el entorno `longtable`, es que nos permite escoger la alineación de la tabla, como si de un entorno flotante se tratara. El parámetro que controla la alineación es el parámetro `alineacion`, que se configura con tres variables: `c` que alinea al centro, `l` que alinea a la izquierda y `r` que alinea a la derecha. Por defecto la tabla viene alineada al centro. Veamos un ejemplo de una tabla completa alineada a la derecha:

Ejemplo 1

Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto

Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto

```

\begin{longtable}[r]{|c|c|}
  \hline
  Texto & Texto\\
  \hline
  Texto & Texto\\
  \hline
  ...
  ...
\end{longtable}

```

3.9.2. Tablas con encabezados y pies en todas las páginas

Una particularidad de este paquete es que nos permite crear tablas con encabezado para la primera página, un encabezado para el resto de las páginas, un pie para la última página y un pie para las demás páginas. Veamos su esquema:

```

\begin{longtable}[alineacion]{columnas}
  \hline
  Contenido del encabezado para la primera pagina\\
  \endfirsthead

  \hline
  Contenido del encabezado para el resto de paginas\\
  \endhead

  Contenido del pie para la ultima pagina\\
  \hline
  \endlastfoot

  Contenido del pie para el resto de paginas\\
  \hline
  \endfoot

  \hline
  Contenido de la tabla\\
  \hline
\end{longtable}

```

Después de definir los encabezados y los pies de la tabla se escribe la tabla como si se tratara de cualquier otra. Para entenderla mejor veamos un ejemplo:

Ejemplo 1

Encabezado para la primera pagina	
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Pie para el resto de paginas	

Encabezado para el resto de paginas	
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Texto	Texto
Pie para la ultima pagina	

```
\begin{longtable}{|c|c|}
\hline
\multicolumn{2}{|c|}{Encabezado para la primera pagina}\\
\endfirsthead
```

```

\hline
\multicolumn{2}{|c|}{Encabezado para el resto de paginas}\\

\endhead

\multicolumn{2}{|c|}{Pie para la ultima pagina}\\
\hline
\endlastfoot

\multicolumn{2}{|c|}{Pie para el resto de paginas}\\
\hline
\endfoot

\hline
Texto & Texto\\
\hline
Texto & Texto\\
\hline
...
...
\end{longtable}

```

3.10. Otros

3.10.1. División de celda de nombres de series

Algo que en la mayoría de las tablas de dos entradas se suele hacer es dividir la primera fila y columna en dos para escribir los nombres de las series. Esto se puede hacer con el paquete `\usepackage{slashbox,pict2e}` usando el comando `\backslashbox{abajo}{arriba}`. Veamos:

Ejemplo 1

Vehiculo \ Color	Color	
	Azul	Rojo
Moto	2	6
Carro	4	10
Avion	6	1

```
\begin{tabular}{|c|c|c|}
\hline
\backslashbox{Vehiculo}{Color} & Azul & Rojo\\
\hline
Moto & 2 & 6\\
\hline
Carro & 4 & 10\\
\hline
Avion & 6 & 1\\
\hline
\end{tabular}
```

3.10.2. Leyenda y citación

A la hora de citar una tabla debe contener una leyenda y una referencia para posteriormente ser citada. Para hacer esto se debe crear la tabla dentro del entorno `table`:

Ejemplo 1

Vehiculo \ Color	Color	
	Azul	Rojo
Moto	2	6
Carro	4	10
Avion	6	1

Cuadro 6: Leyenda de la tabla

```
\begin{table}[!htb]
\centering
\begin{tabular}{|c|c|c|}
\hline
\backslashbox{Vehiculo}{Color} & Azul & Rojo\\
\hline
\end{tabular}
\end{table}
```

```

        \hline
        Moto & 2 & 6\\
        \hline
        Carro & 4 & 10\\
        \hline
        Avion & 6 & 1\\
        \hline
    \end{tabular}
    \caption{Leyenda de la tabla}
    \label{tab:Ve-Co}
\end{table}

```

4. Matemáticas

4.1. Modo matemático en línea

Las ecuaciones en línea son las que hacen parte de un párrafo, insertándose entre texto normal. Para escribir en modo matemático dentro de una línea debemos poner lo deseado entre los signos $\backslash($ y $\backslash)$. Antiguamente se hacía lo mismo con los signos $\$$. Actualmente se puede seguir haciendo, pero lo correcto es hacer uso de los signos $\backslash($ y $\backslash)$.

Ejemplo 1

Texto Texto Texto $y = mx + b$ Texto Texto Texto

```
Texto Texto Texto \ (y=mx+b\ ) Texto Texto Texto
```

Ejemplo 2

Texto Texto Texto $y = mx + b$ Texto Texto Texto

```
Texto Texto Texto $y=mx+b$ Texto Texto Texto
```

4.2. Modo matemático entre párrafos

Los signos $\backslash[$ y $\backslash]$, y $\$$ nos permiten entrar en modo matemático, pero entre párrafo, centrando lo que se encuentre dentro de los mismos:

Ejemplo 1

Texto Texto Texto

$$y = mx + b$$

Texto Texto Texto

```
Texto Texto Texto \ [y=mx+b\ ] Texto Texto Texto
```


Ejemplo 2

Texto Texto Texto

$$y = mx + b$$

Texto Texto Texto

```
Texto Texto Texto $$y=mx+b$$ Texto Texto Texto
```

4.2.1. Entorno equation

Este entorno nos permite enumerar la ecuación que insertemos. Esta quedará entre párrafos y estará centrada. A la derecha de ecuación nos pondrá del número de esta. Es importante tener en cuenta que este entorno solo nos permite escribir una ecuación a la vez.

Ejemplo 1

$$y = mx + b \tag{1}$$

$$a = bh \tag{2}$$

```
\begin{equation}
  y=mx+b
\end{equation}

\begin{equation}
  a=bh
\end{equation}
```

4.3. Escritura matemática

4.3.1. Símbolos

Algunos símbolos en Latex deben ser escritos de forma especial, mientras que otros se comportan como letras, por lo que solo deben ser escritos normalmente. Veamos a continuación algunas tablas con los símbolos más comunes, aunque si se quiere ver más símbolos, recomiendo dirigirse al siguiente link: [RinconMatematico](#)

Símbolos directos				
+	-	=	!	/
()	[]	<
>		,	:	*

Símbolos indirectos			
Símbolo	Comando	Símbolo	Comando
\pm	<code>\pm</code>	\mp	<code>\mp</code>
\times	<code>\times</code>	\neq	<code>\neq</code>
\neg	<code>\neg</code>	∞	<code>\infty</code>
\cap	<code>\bigcap</code>	\cup	<code>\bigcup</code>
\wedge	<code>\bigwedge</code>	\vee	<code>\bigvee</code>
\rightarrow	<code>\rightarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\approx	<code>\approx</code>	\equiv	<code>\equiv</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>
\angle	<code>\angle</code>	\vec{x}	<code>\vec{x}</code>
\forall	<code>\forall</code>	\in	<code>\in</code>
\exists	<code>\exists</code>		

4.3.2. Espacios

Para insertar un espacio cuando se está en modo matemático se deben hacer uso de los siguientes comandos:

Comando	Ejemplo
Normal	$y = mx + b$
<code>\;</code>	$y = m x + b$
<code>\:</code>	$y = m x + b$
<code>\,</code>	$y = m x + b$

4.4. Paquete `amsmath`

El paquete `amsmath` nos permite ampliar las opciones, los entornos y soluciona algunos errores del modo matemático que tiene Latex por defecto. Además agrega símbolos adicionales para usar en los entornos matemáticos. Estos se pueden consultar en la siguiente página: [Günter Milde](#)

Veamos algunas de las opciones que agrega este paquete:

4.4.1. Entorno `equation*`

Este entorno es igual que el entorno `equation`, con la única diferencia de que este entorno no numera las ecuaciones.

Ejemplo 1

$$y = mx + b$$

$$a = bh$$

```
\begin{equation*}
  y=mx+b
\end{equation*}

\begin{equation*}
  a=bh
\end{equation*}
```

4.4.2. Entorno split

Este entorno nos permite alinear diferentes ecuaciones. Este entorno debe de hacerse dentro del entorno `equation` o `equation*` para funcionar. Si se selecciona el entorno `equation`, solo se va a asignar un número para todas las ecuaciones que se hagan. Este entorno es recomendado para los casos en los que se quiere despejar una ecuación.

Este entorno tiene una disposición similar a la de una tabla. Los saltos de línea de deben hacer con un `\\`, y la ecuación debe separarse por `&`, que sirve para establecer el punto donde una ecuación debe ser alineada. El signo que vaya después de `&` será el punto donde las ecuaciones se alineen:

Ejemplo 1

$$\begin{aligned}x + 4 - y + 10 + 2y &= 2x \\14 - y + 2y &= 3x \\y &= 3x - 14\end{aligned}\tag{3}$$

```
\begin{equation}
  \begin{split}
    x+4-y+10+2y &= 2x\\
    14-y+2y &= 3x\\
    y &= 3x-14\\
  \end{split}
\end{equation}
```

Ejemplo 2

$$\begin{aligned}x + 4 - y + 10 + 2y &= 2x \\14 - y + 2y &= 3x \\y &= 3x - 14\end{aligned}$$

```
\begin{equation*}
  \begin{split}
```

```

x+4-y+10+ &2y =2x\\
14-y+ &2y =3x\\
&y =3x-14\\
\end{split}
\end{equation*}

```

4.4.3. Entorno multline

Este entorno está pensado para escribir ecuaciones que son tan largas que no pueden estar en una sola línea. Este paquete parte la ecuación en dos partes separadas por un `\\`. La primera parte se alinea a la izquierda y la segunda a la derecha. Si se separa en más de dos partes las partes intermedias se alinearán al centro. De igual manera existe un entorno que enumera y otro que no lo hace, siendo `multline` y `multline*` respectivamente. Veamos ejemplos de este entorno:

Ejemplo 1

$$p(x) = 3x^6 + 14x^5y + 590x^4y^2 + 19x^3y^3 - 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3 \quad (4)$$

```

\begin{multline}
p(x) = 3x^6 + 14x^5y + 590x^4y^2 + 19x^3y^3\\
- 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3
\end{multline}

```

Ejemplo 2

$$p(x) = 3x^6 + 14x^5y + 590x^4y^2 + 19x^3y^3 - 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3$$

```
\begin{multline*}
p(x) = 3x^6 + 14x^5y\\
+ 590x^4y^2 + 19x^3y^3\\
- 12x^2y^4 - 12xy^5\\
+ 2y^6 - a^3b^3
\end{multline*}
```

4.4.4. Entorno align

Este entorno se encarga de alinear las ecuaciones que estén escritas donde de ella. Este entorno funciona muy similar al entorno `split`, salvo que este no debe ser hecho dentro de un entorno `equation` y permite enumerar varias ecuaciones. Además, este formato permite alinear varias ecuaciones en columnas diferente. Finalmente cuenta con su versión sin enumeración `aling*`.

Las ecuaciones se deben separar con un `\\` y los caracteres escogidos para alinear las ecuaciones deben ser acompañados de un `&`:

Ejemplo 1

$$y = mx + b \tag{5}$$

$$mx + b = y \tag{6}$$

$$x = -b \pm \frac{\sqrt{b^2 - 4ac}}{2a} \tag{7}$$

```
\begin{align}
y &= mx+b\\
mx+b &= y\\
x &= -b \pm \frac{\sqrt{b^2-4ac}}{2a}
\end{align}
```

Ejemplo 2

$$y = mx + b \qquad a = \pi \cdot r^2 \qquad (8)$$

$$mx + b = y \qquad \frac{b \cdot h}{2} = a \qquad (9)$$

$$x = -b \pm \frac{\sqrt{b^2 - 4ac}}{2a} \qquad h^2 = c_1^2 + c_2^2 \qquad (10)$$

```
\begin{align}
  y &= mx+b & a &= \pi \cdot r^2\\
  mx+b &= y & \frac{b \cdot h}{2} &= a\\
  x &= -b \pm \frac{\sqrt{b^2-4ac}}{2a} & h^2 &= c_1^2+c_2^2
\end{align}
```

4.4.5. Entorno gather

Este entorno nos permite centrar una serie de ecuaciones, sin alineación alguna de cierto carácter. Las ecuaciones deben ser separadas con un `\\`. De igual manera al adicionar un asterisco se va a dejar de numerar las ecuaciones.

Ejemplo 1

$$\begin{gathered} y = mx + b \\ mx + b = y \\ x = -b \pm \frac{\sqrt{b^2 - 4ac}}{2a} \end{gathered}$$

```
\begin{gather*}
  y=mx+b\\
  mx+b=y\\
  x=-b \pm \frac{\sqrt{b^2-4ac}}{2a}
\end{gather*}
```

4.5. Paquete amssymb

Agrega símbolos extra y dos tipos de fuentes matemáticas. Las dos fuentes nuevas son `\mathfrak{}`, que es fuente Fraktur y `\mathbb{}` que es negrita de pizarra. De igual manera agrega muchos símbolos que pueden ser consultados en la siguiente página: [Günter Milde](#)

Ejemplo 1

$$\mathfrak{R}$$
$$\mathbb{R}$$
$$\curlyeqsucc$$

```
\begin{gather*}
\mathfrak{F} \\
\mathbb{R} \\
\curlyeqsucc
\end{gather*}
```

4.6. Matrices

Hay que recalcar que para escribir matrices es necesario el uso del paquete `amsmath`. Para escribir una matriz debemos usar en entorno `matrix`, y al igual que las tablas, separamos las columnas con el signo `&` y se da un salto de fila con un signo `\\`. Este entorno debe de usarse dentro de un entorno matemático, como `equation`:

Ejemplo 1

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$$

```
\begin{equation*}
\begin{matrix}
1 & 2 \\
\end{matrix}
\end{equation*}
```



```

      3 & 4 \\
    \end{matrix} \\
\end{equation*}

```

4.6.1. Delimitantes personalizados

Para definir los delimitantes tenemos diferentes opciones. Lo que se debe hacer es cambiar el entorno, manteniendo igual el contenido de este:

Tipo	Entorno	Ejemplo
Plano	<code>\begin {matrix}</code>	$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$
Paréntesis	<code>\begin {pmatrix}</code>	$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$
Paréntesis cuadrados	<code>\begin {bmatrix}</code>	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Corchetes	<code>\begin {Bmatrix}</code>	$\begin{Bmatrix} 1 & 2 \\ 3 & 4 \end{Bmatrix}$
Líneas	<code>\begin {vmatrix}</code>	$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$
Líneas dobles	<code>\begin {Vmatrix}</code>	$\begin{Vmatrix} 1 & 2 \\ 3 & 4 \end{Vmatrix}$

4.6.2. Entorno `smallmatrix` (Matrices en línea)

Si se requiere insertar una matriz en la línea de texto se debe usar el entorno `smallmatrix` dentro del entorno matemático. Además, si se quiere modificar sus delimitantes se deberá usar `\bigl` y `\bigr` para establecer los determinantes. Veamos algunos ejemplos:

Ejemplo 1

Texto Texto Texto Texto Texto Texto $\frac{1}{3} \frac{2}{4}$ Texto Texto Texto Texto Texto Texto

```
Texto Texto Texto Texto Texto Texto \(\begin{smallmatrix} 1 & \\ 2 & \end{smallmatrix} \ 3 & 4 \end{smallmatrix}\) Texto Texto Texto Texto Texto  
Texto
```

Ejemplo 2

Texto Texto Texto Texto Texto Texto $(\frac{1}{3} \frac{2}{4})$ Texto Texto Texto Texto Texto Texto

```
Texto Texto Texto Texto Texto Texto \(\bigl(\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\bigr)\) Texto  
Texto Texto Texto Texto Texto
```

Ejemplo 3

Texto Texto Texto Texto Texto Texto $\{\frac{1}{3} \frac{2}{4}\}$ Texto Texto Texto Texto Texto Texto

```
Texto Texto Texto Texto Texto Texto \(\bigl\{\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\bigr\}\) Texto  
Texto Texto Texto Texto Texto Texto
```

4.7. Creación de símbolos con comandos

Muchas veces necesitamos utilizar símbolos que no tienen un comando asignado por defecto, por lo que los ponemos utilizando diferentes métodos, como tipos de letra, entre otros. Para acortar el tiempo que duramos al poner un símbolo complicado, vamos a crear un comando, o macro, para facilitar esta tarea.

La forma de crear un comando es utilizando `\newcommand{comando}[argumentos][defecto]{definicion}`, esto se debe de posicionar en el preámbulo del documento. En la sección de `comando` se va a definir el nombre que deseamos que tenga nuestro comando. `argumento` es el número de argumentos que le podremos poner al comando de 1 a nueve, como vemos esta es opcional. `defecto` es el argumento que viene por defecto si no se establece un argumento. Y finalmente, `definicion` es donde escribimos lo que hace el comando. Para entender mejor esto veamos varios ejemplos:

Ejemplo 1

```
%Preambulo
\newcommand{\Lagr}{\mathcal{L}}
```

Sin comando: \mathcal{L}

Con comando: \mathcal{L}

```
Sin comando:  $\mathcal{L}$ 
Con comando:  $\mathcal{L}$ 
```

Como vemos en este ejemplo, se creó el comando `\Lagr` que permite hacer el símbolo de Lagrange sin la necesidad de escribir el comando entero.

Ejemplo 2

```
%Preambulo
\newcommand{\R}[1][2]{\mathbb{R}^{\#1}}
\usepackage{amssymb} %Se utiliza este paquete porque incluye el
                     tipo de letra que queremos, es decir, \mathbb{}
```

Sin comando: \mathbb{R}^2

Con comando: \mathbb{R}^2

```
Sin comando:  $\mathbb{R}^2$ 
Con comando:  $\mathbb{R}$ 
```

Como vemos en el ejemplo creamos el comando `\R` que nos permite ilustrar un espacio bidimensional. Como no se especificó un argumento tomó el argumento por defecto, que en este caso es 2. Veamos otro ejemplo donde se especifique el argumento:

Ejemplo 3

```
%Preambulo
\newcommand{\R}[1][2]{\mathbb{R}^{\#1}}
\usepackage{amssymb} %Se utiliza este paquete porque incluye el
    tipo de letra que queremos, es decir, \mathbb{}
```

Sin comando: \mathbb{R}^3

Con comando: \mathbb{R}^3

```
Sin comando: $\mathbb{R}^3$
```

```
Con comando: $\R[3]$
```

Como vemos en este ejemplo, se especificó como argumento el número tres, lo que da el resultado del símbolo para los espacios tridimensionales.

Ejemplo 4

```
%Preambulo
\newcommand{\Cels}{^{\circ}\text{C}}
```

Sin comando: $^{\circ}\text{C}$

Con comando: $^{\circ}\text{C}$

```
Sin comando: ${}^{\circ}\text{C}$
```

```
Con comando: $\Cels$
```

En este ejemplo podemos ver cómo se agrega un comando para escribir el símbolo de los grados Celsius en los entornos matemáticos.