

Apuntes de LaTeX

migueluisV

Actualizadas: Mayo 2023

Índice

1	Comandos básicos y estructura del documento	6
1.1	Lo básico	6
1.2	Unidades y longitud	6
1.3	Títulos, subtítulos y tablas de contenido	7
1.4	Márgenes	9
1.5	Encabezado y pie de página	9
1.5.1	Personalización	10
2	Manipulando el texto del documento	12
2.1	Estilo del texto	12
2.2	Tamaño del texto	12
2.2.1	En todo el documento	12
2.2.2	En algunos bloques de texto	13
2.3	Color del texto	14
2.3.1	Utilizando colores predeterminados	14
2.3.2	Utilizando colores personalizados	15
2.3.3	Color del fondo del texto	15
2.4	Alineación del texto	15
2.5	Interlineado del texto	17
2.5.1	En todo el documento	17
2.5.2	En algunos bloques de texto	17
3	Comentarios	18
4	Escritura de código en \LaTeX	18
5	Referencias y citas	20
5.1	El archivo .bib	20
5.2	Citado de referencias	21
5.3	Sección de referencias en el índice y al final del documento	21
5.4	Estilo de las referencias	21
6	Imágenes	22
7	Listas y secciones	22
7.1	Listas enumeradas	22
7.2	Listas no numeradas	23
7.3	Secciones sin numerar	23
8	Referencias de un objeto en el texto	23

9	Tablas y tabulación	25
9.1	Ubicación inicial y posicionamiento	26
9.2	Referenciar tablas en el texto	30
9.3	Ancho fijo de columnas	30
9.4	Combinar filas y columnas	31
10	Figuras	33
11	Comandos personalizados	33
11.1	Sin parámetros	33
11.2	Con parámetros	34
12	Matemáticas	35
12.1	Fórmulas y área de trabajo de fórmulas	35
12.2	Alineación de ecuaciones: Align, Slip y Multiline	36
12.2.1	Align	36
12.2.2	Slip	36
12.2.3	Multiline	37
12.3	Fracciones	38
12.3.1	Sencillas	38
12.3.2	Fracciones en fracciones	38
12.4	Límites	39
12.5	Raíces	39
12.6	Sumatorias	39
12.7	Integrales	40
12.8	Vectores	40
12.9	Matrices	41
	Referencias	42

Índice de Figuras

1	Imagen de ejemplo	22
2	Mal posicionamiento de tablas, figuras u otros objetos 1	27
3	Mal posicionamiento de tablas, figuras u otros objetos 2	28
4	Figura de ejemplo	33

Índice de Tablas

1	Ejemplo estructura básica de una tabla	26
2	Ejemplo \hline y barras verticales	26
3	Ejemplo \hline y barras verticales	26
4	Ejemplo de tabla con formato APA	30
5	Ejemplo de tabla con ancho de columna fijo	31
6	Ejemplo con tabla con filas combinadas	32
7	Ejemplo con tabla con columnas combinadas	32

Todas la información y comandos usados para crear y ejemplificar este documento fueron sacados de los enlaces puestos en la sección de referencias: (*Bazett, 2022a*), (*Bazett, 2022b*), (*Overleaf, 2022*).

1 Comandos básicos y estructura del documento

Para trabajar con \LaTeX es necesario tener conocimiento de sus nociones básicas y cómo estructurarlo para un mejor orden.

1.1 Lo básico

- **Inicio de comando:** para iniciar un comando en \LaTeX se utiliza una diagonal (\backslash). Por ejemplo:

$\backslash textbackslash$, $\backslash par$, $\backslash LaTeX$

- **Salto de línea:** para hacer un salto de línea, se usan las dos $\backslash\backslash$ o dándo doble ENTER con el teclado (dos salto de líneas).
- **Salto de página:** para hacer un salto de página se usa el comando $\backslash newpage$.
- **Sangría:** Al dejar una línea vacía antes de un párrafo, este tendrá sangría por defecto en \LaTeX . De igual manera, con el comando $\backslash hspace\{ \}$ podemos insertar una sangría a un párrafo, esta sangría debe ser establecida con alguna de las unidades que maneja \LaTeX , que veremos a continuación.
- Un párrafo puede ser creado simplemente escribiendo en el archivo *.tex* después del comienzo del documento, sin embargo, un comando para crear párrafos de una forma más ordenada y con un ancho de párrafo es $\backslash parbox\{ \} \{ \}$, donde el primer par de llaves contiene el ancho del párrafo, y el segundo par contiene el texto de párrafo:

```
\begin{document}
  Esto es un párrafo normal y corriente.

  \parbox{10cm}{Esto es un párrafo con 10cm de ancho}
\end{document}
```

1.2 Unidades y longitud

Para poder establecer el alto o ancho de una imagen, columna, celda, figura, debemos establecer dicha medida de ancho o alto, para ello, requerimos de unidades que \LaTeX conoce y así, establecer el tamaño que deseamos, a continuación está la lista de unidades que se manejan:

- **pt:** puntos, que son aproximadamente 0.3515 milímetros.
- **mm:** milímetros.

- **cm**: centímetros.
- **in**: pulgadas, que son 2.54 centímetros.
- **ex**: el alto de una "x" minúscula dependiendo de la fuente con la que se esté trabajando.
- **em**: el ancho de una "m" mayúscula dependiendo de la fuente con la que se esté trabajando.
- **mu**: su valor es igual a $\frac{1}{18}$ em.
- **sp**: punto especial, 65,536 puntos especiales son iguales a 1pt.

De igual forma, existen algunos comandos de longitud especiales para no rompernos la cabeza utilizando las unidades, los cuales son:

- **\columnwidth**: el ancho de una columna.
- **\paperwidth**: el ancho de toda la página.
- **\paperheight**: el alto de toda la página.
- **\parskip**: el espacio vertical entre párrafos.
- **\textwidth**: el ancho del texto del documento.
- **\textheight**: el alto del texto del documento
- **\topmargin**: el largo del margen superior.

Un ejemplo de la aplicación de los comandos de longitud o unidades de longitud de LaTeX se encuentra a continuación:

```
\usepackage{graphicx}

\begin{document}
  \includegraphics[width=\textwidth]{abc.png}
  \includegraphics[height=5cm]{cba.png}
\end{document}
```

Nota: sea cuidadoso con asignar tamaños a objetos de LaTeX, puede asignarle el tamaño del margen a un objeto y que sobresalga de la página.

1.3 Títulos, subtítulos y tablas de contenido

Títulos y subtítulos

Los títulos y subtítulos en LaTeX dan un orden y estructura al documento que se está trabajando:

- **\section{}**: crea un título o sección.
- **\subsection{}**: crea un subtítulo o subsección.

- `\subsubsection{}`: crea un subtítulo del subtítulo.

```
\begin{document}
  \section{Título 1}
  \subsection{Título 1.1}
  \subsubsection{Título 1.1.1}
\end{document}
```

Despliega:

```
1 Título 1
1.1 Título 1.1
1.1.1 Título 1.1.1
```

Tablas de contenido

Para que se cree correctamente una índice de contenido, ya sea de todo el documento, imágenes, figuras o tablas, es necesario establecer en los puntos donde, se requiere de las secciones, subsecciones y etiquetas de objetos para crear títulos, subtítulos y así sucesivamente, de los índices; por lo general, estos índices suelen colocarse enseguida del inicio del documento. Se utilizan los siguientes comandos:

- Índice:
 - `\tableofcontents`: crea un índice del documento.
 - `\renewcommand*` y `\contentsname`: juntos, cambian el título del índice del documento (que originalmente es "Contents").

```
\begin{document}
  \renewcommand*\contentsname{Índice}
  \tableofcontents
\end{document}
```

- Índice de tablas:
 - `\listoftables`: crea un índice de tablas.
 - `\renewcommand{ }{ }`: cambia el título del índice de tablas.

```
\begin{document}
  \renewcommand{listtablename}{Índice de Tablas}
  \listoftables
\end{document}
```

- Índice de figuras:
 - `\listoffigures`: crea un índice de figuras.
 - `\renewcommand{ }{ }`: cambia el título del índice de figuras.

```
\begin{document}
  \renewcommand{listfigurename}{Índice de Figuras}
  \listoffigures
\end{document}
```


- Índice de ecuaciones: esta opción no está disponible de forma nativa.

1.4 Márgenes

Debemos agregar el paquete **geometry** dentro de nuestro documento para poder modificar los márgenes del mismo.

Entonces, previo a darle comienzo al documento, usamos el comando **newgeometry** y, dentro de sus llaves, agregamos los valores que queremos que tengan nuestros márgenes; los parámetros que puede recibir son: **top**, **bottom**, **outer** e **inner** (*arriba*, *abajo*, *la parte exterior del documento* y *la parte interior*; si estuviéramos viendo el documento como un libro, la parte outer de la página izquierda sería la izquierda, y la parte inner sería el lado derecho, para la página derecha, el outer es el lado derecho y el inner la izquierda de la página; para un documento digital, que se desliza hacia abajo, el outer es el lado derecho y el inner el izquierdo), el valor que se le asigne a estos parámetros tiene que ser un unidades utilizadas por \LaTeX .

```
\usepackage{geometry} % Paquete para trabajar con los márgenes del documento.

% Márgenes del documento.
\newgeometry{
  top=2.5cm, % Superior.
  bottom=2.5cm, % Inferior.
  outer=2.5cm, % Parte exterior.
  inner=2.5cm, % Parte interior.
}

\begin{document}

\end{document}
```

El resultado del ejemplo anterior se puede ver a lo largo de los márgenes de este documento.

1.5 Encabezado y pie de página

\LaTeX por defecto maneja el encabezado y pie de página solamente con el número de página, esto es porque el documento utiliza el comando **\pagestyle{}** que define que es lo que contendrá ambas partes del documento, suele estar ubicado antes del inicio del documento, su estilo por defecto es *plain*, pero se le puede poner otro tipo de estilo, como:

- **plain**: estilo predeterminado de \LaTeX , simplemente agrega el número de página al documento.
- **myheading**: utilizado para documentos con dos lados de página, escribe en la cabecera el título del capítulo en el que se esté posicionado.
- **empty**: deja ambas partes del documento vacío.
- **thispagestyle**: este comando se utiliza para para asignarle un estilo de página a una sola página, esto para diferenciarla de las otras.

`\pagestyle{plain}`

1.5.1 Personalización

Para salirnos de lo básico mostrado en la sección anterior, podemos personalizar aún más el encabezado y pie de página, para ello requerimos el paquete **fancyhdr**, después, dentro del comando `\pagestyle`, le pasamos como parámetro el valor **fancy**, posterior a ello utilizaremos el comando.

`\fancyhf{}`, este último funciona para limpiar todo lo que esté contenido en la cabecera y pie (algo así como empty de `\pagestyle`). La siguiente configuración va previa al comienzo del documento:

```
\usepackage{fancyhdr} % Paquete para personalizar encabezado y pie de página.

% Personalización de la cabecera y pie de página.
\pagestyle{fancy}
\fancyhf{}

\begin{document}

\end{document}
```

Algunos de los parámetros son utilizados en este documento, que se le pueden pasar a **fancyhdr** son:

- **rhead**: texto del lado derecho del encabezado.
- **chead**: texto en el centro del encabezado.
- **lhead**: texto en el lado izquierdo del encabezado.
- **rfoot**: texto en el lado derecho del pie de página.
- **cfoot**: texto en el centro del pie de página.
- **lfoot**: texto en el lado izquierdo del pie de página.

```
\usepackage{fancyhdr} % Paquete para personalizar encabezado y pie de página.

% Personalización de la cabecera y pie de página.
\pagestyle{fancy}
\fancyhf{}
% Texto en esquina superior derecha.
\rhead{Overleaf}
% Texto en esquina superior izquierda.
\lhead{Apuntes de LaTeX}
% Texto en la esquina inferior izquierda.
\lfoot{Hola mundo}
% Texto en esquina inferior derecha (Página n).
\rfoot{Pagina \thepage}

\begin{document}

\end{document}
```

- **headrulewidth**: establece una línea vertical en la cabecera, este comando debe usarse como parámetro para el comando `\renewcommand`. El grosor de esta línea es el parámetro de este comando y puede establecerse con unidades de medida.
- **footrulewidth**: establece una línea vertical en el pie de página, este comando debe usarse como parámetro para el comando `\renewcommand`. El grosor de esta línea es el parámetro de este comando y puede establecerse con unidades de medida.

```
\usepackage{fancyhdr} % Paquete para personalizar encabezado y pie de página.

% Personalización de la cabecera y pie de página.
\pagestyle{fancy}
\fancyhf{}
% Texto en esquina superior derecha.
\rhead{Overleaf}
% Texto en esquina superior izquierda.
\lhead{Apuntes de LaTeX}
% Texto en la esquina inferior izquierda.
\lfoot{Hola mundo}
% Texto en esquina inferior derecha (Página n).
\rfoot{Pagina \thepage}
% Ancho de línea horizontal superior e inferior.
\renewcommand{\headrulewidth}{1pt}
\renewcommand{\footrulewidth}{1pt}

\begin{document}

\end{document}
```

- **Página x de y**: se puede personalizar el estilo en el que se presenta la numeración de páginas, para ello podemos escribir el texto "Página x de z" donde remplazamos x por el comando `\thepage` y z por `\pageref` donde su parámetro es **LastPage** (requiere el paquete **lastpage** para el funcionamiento correcto de esta personalización), con esto obtenemos como resultado que cada página tenga la numeración con el estilo "Página 1 de 120", el estilo de esta presentación puede ser personalizado, lo que importa son los comandos utilizados en este punto, depende de uno escoger si poner este comando en el encabezado o pie.

```
\usepackage{fancyhdr} % Paquete para personalizar encabezado y pie de página.
\usepackage{lastpage} % Paquete para reverenciar páginas del documento.

% Personalización de la cabecera y pie de página.
\pagestyle{fancy}
\fancyhf{}
% Texto en esquina superior derecha.
\rhead{Overleaf}
% Texto en esquina superior izquierda.
\lhead{Apuntes de LaTeX}
% Texto en la esquina inferior izquierda.
\lfoot{Hola mundo}
% Texto en esquina inferior derecha (Página n de n).
\rfoot{Pagina \thepage \hspace{1pt} de \pageref{LastPage}}
```

```
% Ancho de línea horizontal superior e inferior.
\renewcommand{\headrulewidth}{1pt}
\renewcommand{\footrulewidth}{1pt}

\begin{document}

\end{document}
```

Nota: los comandos `\thepage` referencia a la página actual en el contador del documento.

El desarrollo de los ejemplos anteriores se ven a lo largo de todos las cabeceras y pies de página de este documento.

2 Manipulando el texto del documento

La necesidad de agrandar o encoger la fuente en un documento es algo básico, en \LaTeX se puede satisfacer esta necesidad. Los tamaños de letra por defecto de este lenguaje son los de 10pt, 11pt y 12pt para tipos de documentos *article* o *report*, siendo 10pt con el que se crea todo nuevo documento, si intentamos poner algún otro valor con otra unidad, se pondrá automáticamente 10pt, por lo que se tendrá que recurrir a paquetes externos.

2.1 Estilo del texto

Para resaltar textos se usan los siguientes comandos:

- **Negrita:** se usa el comando `\textbf{}` para hacer el texto negrita.
- Subrayado: se usa el comando `\underline{}` para hacer una subrayar el texto.
- *Cursiva:* se usa el comando `\textit{}` para hacer el texto cursiva.

```
\begin{document}
  \textbf{Este texto es negrita.}
  \textit{Este texto es cursiva.}
  \underline{Este texto está subrayado.}
\end{document}
```

2.2 Tamaño del texto

2.2.1 En todo el documento

Previo a mostrar la solución de aumentar o reducir la fuente de un documento con un paquete externo, queremos presentar que hay dos formas de cambiar el tamaño del texto del documento \LaTeX , la primera es hacer que todo el texto de todo el documento cambie de tamaño desde la instrucción `documentclass{}`, como se ve a continuación:

```
% Tipo de documento con tamaño de fuente de 12pt.
% \documentclass[12pt]{article}
```

Nota: el segundo comando del ejemplo anterior fue comentado porque, a pesar de estar en un bloque de código, el comando se sigue ejecutando a pesar de que no debería ejecutarse.

Como podemos apreciar, utilizamos uno de los tamaños de letra predeterminado de \LaTeX entre corchetes previo al tipo de documento, sin embargo, seguimos limitado a los tamaños 10, 11 y 12 puntos.

2.2.2 En algunos bloques de texto

La segunda solución para cambiar el tamaño de la fuente del documento es utilizar las siguientes palabras reservadas o comandos de \LaTeX :

- `\tiny`: tamaño de letra más pequeño.
- `\scriptsize`: tamaño un poco mayor a *tiny*.
- `\footnotesize`: tamaño de letra similar a los textos en pies de página.
- `\small`: tamaño de letra un poco menor al tamaño de letra predeterminado del documento (10pt, 11pt o 12pt).
- `\normalsize`: el mismo tamaño que el tamaño de letra predeterminado del documento (10pt, 11pt o 12pt).
- `\large`: tamaño de letra un poco mayor a *normalsize*.
- `\Large`: tamaño de letra un poco mayor a *large*.
- `\LARGE`: tamaño de letra un poco mayor a *Large*.
- `\huge`: tamaño de letra un poco mayor a *LARGE*.
- `\Huge`: tamaño de letra un poco mayor a *huge* y el tamaño más grande.

Una demostración de estos tamaños en un párrafo es la siguiente:

```
\tiny lorem ipsum.  
\scriptsize lorem ipsum.  
\footnotesize lorem ipsum.  
\small lorem ipsum.  
\normalsize lorem ipsum.  
\large lorem ipsum.  
\Large lorem ipsum.  
\LARGE lorem ipsum.  
\huge lorem ipsum.  
\Huge lorem ipsum.
```

lorem ipsum.

lorem ipsum.

lorem ipsum.

lorem ipsum.

lorem ipsum.

lorem ipsum.
 lorem ipsum.
 lorem ipsum.
 lorem ipsum.
 lorem ipsum.

Si busca aplicar alguno de estos tamaños a solo unas cuantas palabras de un párrafo, siga el siguiente ejemplo:

```
{\Large lorem ipsum lorem ipsum lorem ipsum lorem ipsum} lorem ipsum lorem
ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
ipsum lorem ipsum lorem ipsum lorem ipsum
```

lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum

Sea consiente de que estas instrucciones no le permiten ingresar un tamaño y unidad de medida de su preferencia, si desea realizar dicha acción, debe acudir a paquetes externos, como lo muestra este [video](#).

2.3 Color del texto

Por defecto, \LaTeX no permite utilizar colores para pintar un texto en específico, por lo que tendrá que recurrir a paquetes externos, como lo pueden ser **color** y **xcolor**:

```
\usepackage{color}
\usepackage{xcolor}
```

La principal diferencia entre ambos es que el primer no tiene tantos colores predeterminados por defecto, del mismo modo que no se pueden crear o personalizar colores con tantos modelos (hexadecimal, RGB, RGBA, por ejemplo), mientras que el segundo si tiene estas posibilidad, por lo que recomendamos utilizar el segundo paquete.

2.3.1 Utilizando colores predeterminados

Basta con importar el paquete de su preferencia, para fines de este documento y sección se utilizará *xcolor*, y utilizar el comando **textcolor** para pintar una o varias palabras de otro color:

```
Este \textcolor{red}{mensaje} tiene \textcolor{blue}{varias} palabras \textcolor{green}{en}
} distintos \textcolor{purple}{colores}.
```

```
\textcolor{red}{Este mensaje está pintado todo de un color}.
```

Este **mensaje** tiene **varias** palabras **en** distintos **colores**.

Este mensaje está pintado todo de un color.

2.3.2 Utilizando colores personalizados

Si requiere de un color distinto a los disponibles en los paquetes externos, utilice el comando **definecolor** con los siguientes valores:

$$\backslash definecolor\{nombre\}\{modelo\}\{configuración\}$$

Donde:

- nombre: es el nombre del color que se está creando.
- modelo: es la forma en la que se creará el color, pudiendo ser escala de grises (*gray*), *rgb*, *RGB*, *HTML* o cian, magenta, amarillo y negro (*cmcy*)
- configuración: es la definición o configuración del color. Según el modelo será la configuración, por ejemplo, el modelo HTML utiliza valores hexadecimales como configuración o RGB utiliza un trío de valores para el rojo, verde y azul (0,0,0 por ejemplo).

Este documento tiene definido los siguientes colores:

```
% Define colores nuevos.
\definecolor{color}{HTML}{E4E4EE}
\definecolor{verde}{HTML}{3C8031}
```

Texto con el color llamado 'color'.

Texto con el color llamado 'verde'.

Ambos colores son utilizados más que nada en las secciones de código de este documento. Puede consultar este [enlace](#) o [este otro](#) para consultar más información acerca de la creación de colores con xcolor y sus colores predeterminados disponibles.

2.3.3 Color del fondo del texto

Siguiendo con la línea de colorear objetos en documentos L^AT_EX, también se puede pintar el fondo de un párrafo o texto utilizando el comando **colorbox**, en conjunto con el uso de colores que se ha explicado previamente.

```
Esta \colorbox{orange}{palabra} tiene un color de fondo naranja.
```

Esta palabra tiene un color de fondo naranja.

2.4 Alineación del texto

Por defecto, L^AT_EX utiliza la alineación de texto **justificada**, si es necesario alinear nuestro texto a la izquierda, centrado o derecha, se tienen las siguientes opciones.

A la izquierda

Utilice los comandos **begin{flushleft}** y **end{flushleft}** para alinear un texto a la izquierda:

```
\begin{flushleft}
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum} lorem ipsum lorem ipsum
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
  ipsum lorem ipsum lorem ipsum lorem ipsum
\end{flushleft}
```

lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum

Centrado

Visto anteriormente para alinear tablas, figuras o mensajes importantes, el comando **`begin{center}`** y **`end{center}`** alineará su texto al centro:

```
\begin{center}
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum} lorem ipsum lorem ipsum
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
  ipsum lorem ipsum lorem ipsum lorem ipsum
\end{center}
```

lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum

La diferencia entre el comando anterior y **`centering`** es que el primero tiene un inicio y fin, mientras que el segundo terminará su efecto hasta que aparezca otro comando que lo interrumpa, es esta razón por la cual se recomienda utilizar *centering* en figuras o tablas, mientras que *center* es recomendado para textos.

A la derecha

Utilice los comandos **`begin{flushright}`** y **`end{flushright}`** para alinear un texto a la izquierda:

```
\begin{flushright}
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum} lorem ipsum lorem ipsum
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
  lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
  ipsum lorem ipsum lorem ipsum lorem ipsum
\end{flushright}
```

lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
 lorem ipsum lorem ipsum lorem ipsum

Consulte [este enlace](#) para tener más información sobre la alineación de texto u las alternativas.

2.5 Interlineado del texto

El interlineado en \LaTeX no es algo que le interese a los usuarios, ya que este lenguaje se encarga de ajustar el interlineado al tipo de documento que se esté trabajando, sin embargo, siempre existirá la duda de cómo ajustarlo a nuestra preferencia. Para lograr este cometido, es necesario reescribir el comando que se encarga de ajustar el interlineado de todo el documento. El interlineado actual de este documento es sencillo, por lo que puede tomar estos ejemplos y aplicarlos a su documento.

2.5.1 En todo el documento

El comando **baselinestrech** es el que se encarga de establecer el interlineado de un documento, por defecto, el valor de este comando es 1, veamos como se utiliza.

```
% Sobre escribe el comando baselinestrech con el valor 2.
\renewcommand{\baselinestrech}{2}

% \begin{document}
% Inicio del documento...
% \end{document}
```

Se utiliza el comando *renewcommand* para sobre escribir un comando anterior, en este caso, *baselinestrech*, el valor que se le puede poner puede ser 1, 1.5, 2, o alguno otro conocido o utilizado en otros editores de texto. Puede consultar este [enlace](#) para más información referente al interlineado.

2.5.2 En algunos bloques de texto

Agregamos el paquete **setspace**, el cual posee los comandos para establecer un interlineado simple (1), de 1.5 y doble (2), con la posibilidad de crear un interlineado propio, de 1.8 por ejemplo.

$$\backslash usepackage{setspace}$$

```
\doublespacing
Todo este texto tendrá interlineado doble
\onehalfspace
Todo este texto tendrá interlineado uno y medio
\singlespace
Todo este texto tendrá interlineado sencillo
\spacing{1.6}
Todo este texto tendrá interlineado de 1.6
```

Como podemos apreciar, este comando no tiene un final, por lo que después del mismo todo el texto se verá afectado por el interlineado establecido hasta que aparezca otro comando que lo interrumpa. Puede revisar este [enlace](#) para consultar un poco más de información.

3 Comentarios

Para poner un comentario se usa el símbolo %:

% Esto es un comentario.

Si lo que se busca es comentar varias líneas de código al mismo tiempo sin ir una a una, podemos utilizar el paquete **comment** y utilizar su comando **comment**:

```
\usepackage{verbatim} % Paquete para comentar varias líneas a la vez.

\begin{document}
  \begin{comment}
    esto
    es
    un
    bloque
    de
    texto
    comentado
  \end{comment}

  hola mundo
\end{document}
```

Si buscamos el camino más rápido, Overleaf (la plataforma online donde actualmente estamos escribiendo este documento) proporciona el *shortcut* **Ctrl** + **/** para comentar varias líneas de texto ya seleccionadas con el cursor. Este *shortcut* puede variar dependiendo el IDE o editor de texto donde se esté desarrollando el proyecto **L^AT_EX**.

4 Escritura de código en **L^AT_EX**

Algo muy interesante a la hora de trabajar en este ambiente, es el lograr escribir código de programación que se vea elegante y tabulado para su mejor comprensión, esto lo logramos con el paquete **listing**.

```
\usepackage{listings}
```

Lo más difícil (o sencillo), es personalizar o configurar este paquete para que se mire como nosotros deseamos, para ello utilizamos el comando **lstset{}**, este debe ser posicionado antes del comienzo del documento, y dentro de sus llaves podemos insertar muchas entradas o atributos que configurarán el resultado final del paquete. Algo que quiero mencionar es que este paquete soporta automáticamente la escritura y estilo de varios lenguajes de programación (Python, Java, C, C++ o HTML, lamentablemente CSS y JavaScript no están disponibles), esto es un tipo de entrada o atributo que podemos configurar en *lstset*, deajo a continuación la configuración usada en este documento junto con algunos otros atributos que podrían serle útiles para el Caso 1, y otra que podría utilizarse para escribir código de CSharp en el Caso 2:

Caso 1, configuración completa de este archivo:

```

\usepackage{xcolor} % Paquete básico para agregar color al texto.

% Define colores nuevos.
\definecolor{color}{HTML}{E4E4EE}
\definecolor{verde}{HTML}{3C8031}

% Personalización de la fuente para el código.
\lstset{
  language = TeX, % Lenguaje con palabras reservadas de este resaltadas.
  basicstyle = \ttfamily\footnotesize,% Utiliza la fuente ttfamily, en especial el
    paquete inconsolata.
  frame = single, % Quita el marco al cuadro flotante que contiene el código o texto.
  backgroundcolor = \color{color}, % Cambia el color del fondo del marco del código.
    Utiliza el paquete "xcolor" y define un nuevo color.
  columns = fullflexible, % Ajusta el cuadro flotante al tamaño del texto del documento.
  breaklines = true, % Ajusta el texto dentro del contenedor.
  inputencoding = utf8, % Admite caracteres del código UTF8.
  extendedchars = true, % Soporte para caracteres especiales.
  %numbers = left, % Agrega número de línea al código (izquierda, sin número y derecha).
  showstringspaces = false, % Quita los guiones bajos predeterminados de los espacios en
    cadenas de texto.
  escapebegin = \obeyspaces, % Complemento de la entrada anterior.
  % rulecolor = \color{red}, % Color del borde del marco del código.
  % numberstyle = \color{red}, % Color de los números en el texto o código.
  % stringstyle = \color{red}, % Color de las cadenas de texto en el texto o código.
  % keywordstyle = \color{red}, % Color de las palabras reservadas en el texto o código.
  % identifierstyle = \color{red}, % Color del texto o código.
  commentstyle = \color{verde}, % Color de los comentarios en el texto o código.
  literate = % Acepta los siguientes caracteres especiales fuera de UTF8.
    {á}{\a}1 {é}{\e}1 {í}{\i}1 {ó}{\o}1 {ú}{\u}1
    {Á}{\A}1 {É}{\E}1 {Í}{\I}1 {Ó}{\O}1 {Ú}{\U}1
    {ñ}{\n}1 {Ñ}{\N}1,
}

```

Puedes agregar el comando `\footnotesize` para hacer un poco más pequeña la letra del código, esto va justo enseguida de `\ttfamily` en la entrada `basicstyle`.

Caso 2, configuración rápida con lenguaje:

```

\lstdefinestyle{sharpc}{language=[Sharp]C, frame=lr, rulecolor=\color{blue!80!black}}

\lstset{style=sharpc}

```

Para comenzar a trabajarlo utilizamos los comandos `\begin{}` y `\end{}`, dentro de sus llaves escribimos la palabra **lstlisting**:

```

\begin{lstlisting}
Esto es código!!!
\end{lstlisting}

```

En el Caso 1, todos los bloques *lstlisting* tendrán el mismo aspecto (en especial cuando se establece un tipo de lenguaje para el estilo de los bloques), mientras que el Caso 2 permite crear varios estilos para distintos lenguajes de programación, o simplemente permite establecer varios estilos o temas para los bloques *lstlisting*.

5 Referencias y citas

Existen tres formas de trabajar con citas y referencias en \LaTeX , en esta ocasión y según mi entendimiento, la más sencilla es la de **biblatex**.

5.1 El archivo .bib

Referenciar y citar por medio de *biblatex* requiere que tengamos a la mano un archivo con extensión *.bib* dentro del directorio donde estemos trabajando el proyecto, ahí estarán escritas todas las referencias que usaremos a lo largo del escrito.

Una vez creado el archivo *.bib* con las referencias escritas, necesitamos integrarlo a nuestro archivo **main**, para ello usamos el comando **addbibresource** en la parte superior del mismo (junto con los otros paquetes que estemos utilizando), y dentro de sus llaves escribimos el nombre del archivo *.bib* que se creó anteriormente.

```
\addbibresource{referencias.bib}
```

Para escribir las fuentes dentro de este archivo, existe una categoría para cada tipo de fuente, es decir, las fuentes están categorizadas por el tipo *online*, *artículo*, *libro*, *revista*, *artículo científico*, etc. Cada categoría puede tener un tipo de entrada, dato o atributo en la cual podemos insertar la información de nuestra fuente (*autor*, *fecha de publicación*, *link*, *página*, *volumen*, etc), es necesario investigar el tipo de entrada o atributo posee cada fuente que necesitamos, ya que no todas las categorías aceptan todas o ciertas entradas o atributos (no tiene sentido que la categoría *online* tenga una entrada tipo *volumen*). A continuación, se presentan tres ejemplos de categorías de fuentes con sus entradas correspondientes:

```
@online{trefor1,
  title = "Intro to LaTeX : Learn to write beautiful math equations",
  author = "Trefor Bazett",
  date = {2022-02-15},
  url = "https://youtu.be/Jp0lPj2-DQA",
  publisher = "[Video de YouTube]"
}

@online{trefor2,
  author = "Trefor Bazett",
  title = "How to write a thesis using LaTeX **full tutorial**",
  date = {2022-02-15},
  url = "https://youtu.be/zqQM66uAig0",
  publisher = "[Video de YouTube]"
}

@online{overl,
  author = "Overleaf",
  title = "Overleaf Help",
  date = {2022-02-15},
  url = "https://www.overleaf.com/learn",
  publisher = "[Video de YouTube]"
}
```

Nota: puedes acceder al sitio de Scribbr para escribir referencias y citas APA, este sitio te ofrece exportar las referencias APA a bibLaTeX.

5.2 Citado de referencias

Ya con el archivo de referencias creado y adjuntado a nuestro fichero **main**, es momento de citar nuestras referencias, para lograr esto usamos el comando `\cite{}`, dentro de sus llaves escribimos la palabra clave o el nombre que le dimos previamente a cada una de nuestras fuentes (en el ejemplo anterior, los nombres de las referencias son: *trefor1*, *trefo2* y *overl*).

`\cite{trefor1}`

5.3 Sección de referencias en el índice y al final del documento

Para crear la sección de referencias se usa el comando `\printbibliography`, por si solo cumple con su función, pero si se busca cambiar la palabra por defecto que tiene (References), se pueden agregar unos corchetes después del comando donde se le pasa el parámetro *title* para cambiar el título de la sección, el valor de este parámetro debe estar contenido dentro de llaves; para que esta sección salga en el índice general del documento sin estar enumerada, dentro de los mismos corchetes donde está el parámetro *title*, le pasamos otro parámetro separado por una coma llamado *heading*, su valor contenido dentro de llaves sería *bibintoc*.

```
\usepackage{biblatex} % Paquete para usar referencias y citas.
\addbibresource{referencias.bib} % Se agrega el archivo .bib al documento.

\begin{document}
  \renewcommand*{\contentsname}{Índice}
  \tableofcontents
  \printbibliography[title={Referencias}, heading={bibintoc}]
\end{document}
```

El resultado del ejemplo anterior lo puede revisar en este mismo documento.

5.4 Estilo de las referencias

Si queremos aplicar un estilo en particular a las referencias, es posible modificar el que está puesto por defecto para que nuestras citas y sección de referencias tengan otro aspecto (APA, IEEE, entre otros); para este caso en particular yo usaré el estilo APA para este documento.

Cuando agregamos el paquete **biblatex**, no le pasamos ningún otro parámetro a *usepackage*, ahora, entre corchetes, antes de las llaves, le pasaremos el parámetro **style**, y su valor será igual a **apa**, repito, en esta parte del comando se le puede usar otro tipo de estilo, y automáticamente, las citas y la sección de referencias se actualizará al etilo establecido.

```
\usepackage[style=apa]{biblatex} % Paquete para usar referencias y citas.
\addbibresource{referencias.bib} % Se agrega el archivo .bib al documento.

\begin{document}

\end{document}
```

6 Imágenes

Para usar imágenes dentro del documento, debemos tener en el mismo el paquete **graphicx** y las imágenes en el directorio del proyecto.

Para adjuntar la imagen o imágenes a un documento se usa el comando siguiente:

`\includegraphics[]{}`

El primer parámetro entre corchetes contiene el tamaño de la imagen y el segundo parámetro la imagen como tal; hay varias formas de indicar el tamaño, como lo puede ser la escala de la imagen, por medio del comando **width** (anchura) o **height** (altura):

```
\usepackage{graphicx} % Paquete para usar imágenes y figuras.  
  
\begin{document}  
  \includegraphics[width=10cm]{ejemplo_latex.png}  
\end{document}
```

Figure 1: Imagen de ejemplo



La *Figura 1* posee la dimensión del ejemplo que se acaba de mostrar. Se recomienda trabajar imágenes de forma centrada usando los comando **`\begin{center}`** y **`\end{center}`** para manejar mejor el comando de inserción de la imagen, su nombre (*caption*) y etiqueta de referencia (*label*).

7 Listas y secciones

7.1 Listas enumeradas

Para crear una lista numerada, se usa el comando **`\begin`** y **`\end`**, seguido de las llaves donde su contenido es **enumerate**, con esto se crea una lista numerada, para cada elemento de esta lista se usa el comando **`\item`**:

```
\begin{enumerate}  
  \item LE1  
  \item LE2  
  \item LE3  
\end{enumerate}
```

1. LE1
2. LE2
3. LE3

7.2 Listas no numeradas

Si no se busca una lista numerada, solo se cambia el contenido dentro de las llaves del comando **begin**, por ejemplo: **itemize**

```
\begin{itemize}
  \item LI1
  \item LI2
  \item LI3
\end{itemize}
```

- LI1
- LI2
- LI3

Para hacer listas dentro de listas, basta con comenzar una nueva lista en uno de los items o debajo del comando `item`.

7.3 Secciones sin numerar

Si se desea que los títulos de secciones o subsecciones no estén numerados, se utiliza el ya utilizado comando `\section{}`, solo que entre el inicio de las llaves y el final de la palabra reservada, se pone un `*`.

`\section*{Prueba de sección sin enumerar}`

Prueba de sección sin enumerar

8 Referencias de un objeto en el texto

Para referenciar un objeto (tabla, imagen o figura) en LaTeX se puede usar el comando `\label{}`, dentro de sus llaves damos nombre al objeto a referenciar. Es importante recalcar que para lograr un referenciado correcto, primero se debe usar el comando `\caption{}` y después el comando `label`, esto para que no haya errores con el contador de referencias. Además, para llevar un orden en el nombre de las referencias dentro de `label`, se puede usar la palabra clave *eq*, para ecuaciones; *tab*, para tablas; *img*, para imágenes y *fig*, para figuras.

```

\caption{Imagen de ejemplo}
\label{img: 1}
\caption{Tabla de ejemplo}
\label{tab: 1}
\caption{Ecuación de ejemplo}
\label{eq: 1}
\caption{Figura de ejemplo}
\label{fig: 1}

```

En este caso, crearé una ecuación con el comando `\begin` y `\end`, dentro de sus llaves va la palabra **equation**, el cual numera automáticamente todas las ecuaciones puestas en el documento, y luego la referenciaré en un texto aparte. Para referenciar a las ecuaciones u objetos en otra parte del documento, se usa el comando `\ref{}`.

```

\begin{equation}
\label{ec: 1}
x^2+2ab+y
\end{equation}

\begin{equation}
\label{ec: 2}
x^2+2ab+y=1
\end{equation}

La Ecuación \ref{ec: e1} posiblemente está mal escrita. \\
La Ecuación \ref{ec: e2} igual.

```

$$x^2 + 2ab + y \tag{1}$$

$$x^2 + 2ab + y = 1 \tag{2}$$

La *Ecuación 1* posiblemente está mal escrita.

La *Ecuación 2* igual.

9 Tablas y tabulación

Para trabajar con tablas, se tiene que usar los comandos `\begin` y `\end`, seguido de `table`: es un indicativo para señalar que se comenzará a trabajar con una tabla, y el comando `tabular` (que tiene el mismo inicio de comando *begin* y *end*) es propiamente la tabla.

El comando `\caption{}` dentro de `table`, es usado para darle un título a la tabla y el comando `label` funciona para referenciar la tabla en un texto o la sección de referencias.

```
\begin{table}
  \caption{Nombre}
  \label{tab: n}
  \begin{tabular}
    & \\\
    & \\
  \end{tabular}
\end{table}
```

Al trabajar con *tabular*, abrimos un par de llaves para ingresar parámetros que indican la alineación del contenido de cada columna (si la tabla tiene tres columnas, cada una puede tener distinta alineación). Por defecto, la tabla no tiene líneas verticales de separación de columnas, estas pueden ser agregadas en dentro de estos parámetros con una línea vertical | entre cada parámetro de alineación; para agregar líneas horizontales de separación de filas a la tabla, se usa el comando `\hline`.

La Tabla 1 solamente trata la estructura básica de una tabla, sin líneas verticales ni horizontales, mientras que la Tabla 2 involucra otra alineación de columnas y todos sus bordes:

```
\begin{table}[H]
  \begin{center}
    \caption{Ejemplo estructura básica de una tabla}
    \label{tab: 1}
    \begin{tabular}{l c r }
      C1 & C2 & C3 \\\
      C4 & C5 & C6 \\
    \end{tabular}
  \end{center}
\end{table}

\begin{table}[H]
  \begin{center}
    \caption{Ejemplo \textbackslash{hline} y barras verticales}
    \label{tab: 2}
    \begin{tabular}{|c|c|c|}
      \hline
      C1 & C2 & C3 \\\
      \hline
      C4 & C5 & C6 \\\
      \hline
    \end{tabular}
  \end{center}
\end{table}
```

Table 1: Ejemplo estructura básica de una tabla

C1	C2	C3
C4	C5	C6

Table 2: Ejemplo `\hline` y barras verticales

C1	C2	C3
C4	C5	C6

Si pone atención, se dará cuenta que las dos diagonales invertidas (`\\`) funcionan para pasar entre fila y fila dentro de una tabla, pero, ¿y si yo quiero dar saltos de línea dentro de una celda?, recordemos que esas mismas diagonales son necesarias para dar un salto en los párrafos.

Podemos incluir el comando anteriormente mencionado: `\parbox{ }{ }` dentro de las celdas para poder dar saltos de línea tranquilamente (*Tabla 3*):

```
\begin{table}[H]
  \begin{center}
    \caption{Ejemplo \textbackslash{hline} y barras verticales}
    \label{tab: 3}
    \begin{tabular}{|c|c|c|}
      \hline
      C1 & C2 & \parbox{3cm}{C3.1 \\ C3.2 \\ C3.3} \\
      \hline
      C4 & C5 & \parbox{3cm}{C6.1 \\ C6.2 \\ C6.3} \\
      \hline
    \end{tabular}
  \end{center}
\end{table}
```

Table 3: Ejemplo `\hline` y barras verticales

C1	C2	C3.1 C3.2 C3.3
C4	C5	C6.1 C6.2 C6.3

9.1 Ubicación inicial y posicionamiento

Las tablas suelen ser ubicadas del lado izquierdo cuando son insertadas, podemos usar los comandos `\begin{center}` y `\end{center}` o `\centering` para centrarlas en el documento, personalmente recomiendo *center* porque es más fácil de controlar.

```
\begin{center}
  texto
\end{center}

\centering
```

El posicionamiento de tablas en el texto del documento es otro punto a considerar, esto aplica para tablas, ecuaciones y figuras, pero lo trataremos específicamente para tablas.

En un editor de texto, como sería Word, insertamos una tabla en la posición actual del cursor, esto puede ser al inicio o final del documento, o después de un párrafo, y la tabla se posicionaría en el documento según cómo se mueva el párrafo, imagen u otro objeto previo a la tabla.

LaTeX por otra parte, pone las tablas en la parte superior de la página donde quedó el último párrafo, imagen, figura u objeto por defecto, esto puede ocasionar problemas si un texto que referencia a la tabla queda en una página distinta o no conviene el posicionamiento de la tabla con respecto al contenido del documento. El vistazo a lo que nos referimos se encuentra en las *Figuras 2 y 3*:

Figure 2: Mal posicionamiento de tablas, figuras u otros objetos 1

La Ecuación 1 posiblemente está mal escrita.

La Ecuación 2 igual.

Página 15 de 22

Table 1: Ejemplo estructura básica de una tabla

C1	C2	C3
C4	C5	C6

8 Tablas y tabulación

Para trabajar con tablas, se tiene que usar los comandos `\begin` y `\end`, seguido de `table`: es un indicativo para señalar que se comenzará a trabajar con una tabla, y el comando `tabular` (que tiene el mismo inicio de comando *begin* y *end*) es propiamente la tabla.

El comando `\caption{}` dentro de `table`, es usado para darle un título a la tabla y el comando `label` funciona para referenciar la tabla en un texto o la sección de referencias.

Figure 3: Mal posicionamiento de tablas, figuras u otros objetos 2

```

\end{tabular}
\end{center}
\end{table}

```

8.1 Ubicación inicial y posicionamiento

Las tablas suelen ser ubicadas del lado izquierdo cuando son insertadas, podemos usar los coman-

Página 16 de 22

Table 2: Ejemplo \hline y barras verticales

C1	C2	C3
C4	C5	C6

dos `\begin{center}` y `\end{center}` o `\centering` para centrarlas en el documento, personalmente recomiendo `center` porque es más fácil de controlar.

Las tablas que vemos en las imágenes anteriores son las que ya pusimos previamente, pero con un posicionamiento por defecto. Vemos que la Figura 2 posiciona la tabla al principio de la página 15, previo a la sección de Tablas y tabulación; la Figura 3 posiciona otra tabla al principio de la siguiente página, no donde nosotros lo deseamos.

Podemos pasar un parámetro de posicionamiento entre corchetes a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ dónde queremos la tabla, en todas las tablas usadas en este documento se utiliza el parámetro H , esto debido a que permite que la tabla se ponga justo a continuación del último objeto o texto donde estuvimos posicionados, respetando el espacio de los objetos o textos a continuación. A continuación se muestra la lista de parámetros de posicionamiento:

1. **h**: pone la tabla aproximadamente aquí.
2. **p**: pone la tabla en una página especial, solo para tablas.
3. **t**: pone la tabla en la parte superior de la página.
4. **b**: pone la tabla en la parte inferior de la página
5. **!**: fuerza el posicionamiento de la tabla en base a los parámetros listados aquí.
6. **H**: pone la tabla exactamente aquí. Requiere del paquete **float** para poder ser utilizado.

```

\begin{document}
% Parámetro entre corchetes ([]) con el valor H para

```

```
% posicionar estrictamente en dicha posición al
% objeto.
\begin{table}[H]

\end{table}
\end{document}
```

Nota: personalmente recomiendo el parámetro H, posiciona la tabla donde queremos y respeta el espacio del resto del contenido del documento.

En ocasiones, cuando se está escribiendo un documento y se está trabajando con tablas, estas pueden chocar con el posicionamiento o espacio de las imágenes, por ejemplo, si tenemos una imagen, en seguida una tabla y luego una nueva sección o subsección, en vez de seguir ese orden, puede que ocurra que aparezca la imagen, la sección y luego la tabla, cosa que no va de acuerdo al orden que buscamos, para solucionar eso, puede insertar el comando `\newpage` entre el fin de la imagen y el inicio de la tabla para lograr acomodar las cosas.

Otra solución es utilizar el parámetro entre corchetes ([]) con el valor H para las figuras, imágenes o ecuaciones:

```
\begin{document}
  Imagen
  \begin{figure}[H]

\end{figure}
  Tabla
  \begin{table}[H]

\end{table}
\end{document}
```

De esta manera, ambos objetos respetan su propio posicionamiento y espacio, sin chocar ni afectar al resto de objetos o textos.

```
\begin{table}[H]
\begin{center}
\caption{Ejemplo de tabla con formato APA}
\label{tab: 4}
\begin{tabular}{l l l l}
\hline
Título 1 & Título 2 & Título 3 & Título4 \\\hline
aaaaaaaaaa & bbbbbbbbbbbb & cccccccc & lorem ipsum \\\hline
prueba 1 & prueba 2 & prueba tres & prueba four \\\hline
texto de relleno & relleno & info adicional & oh yeah \\\hline
\end{tabular}
\end{center}
\end{table}
```

Table 4: Ejemplo de tabla con formato APA

Título 1	Título 2	Título 3	Título4
aaaaaaaaaaaa	bbbbbbbbbbbbbb	cccccccc	lorem ipsum
prueba 1	prueba 2	prueba tres	prueba four
texto de relleno	rellenando	info adicional	oh yeah

La Tabla 4 tiene un formato parecido a lo establecido al formato APA, esto es un ejemplo de como trabajar de forma sencilla con las tablas.

9.2 Referenciar tablas en el texto

Para poder referenciar en texto una tabla, es recomendable poner el texto de referencia dentro del comienzo de la tabla y previo del comienzo de **tabular**, antes de centrarla si es que se quiere centrar, para que, si se mueve la tabla, el texto de referencia se vaya con ella también, ya que si se pone afuera, el texto puede no quedar junto o cerca a la tabla.

9.3 Ancho fijo de columnas

Para que una tabla tenga columnas con un ancho fijo, se utiliza el paquete **array**.

`\usepackage{array}`

El tamaño del ancho de columna se indica dentro de llaves en las llaves donde se indica la alineación del contenido de las columnas, se tiene que tener conocimiento sobre las unidades de medida de \LaTeX .

`\begin{tabular}{m{5cm} m {3cm} m{1cm}}`

En esta ocasión, como podemos recordar de los parámetros de las tablas convencionales (l, c, r), en este caso dejamos de lado los parámetros de alineación del contenido de columnas y los sustituiremos por los siguientes:

- **m**: alinea el texto en medio de la columna.
- **b**: alinea el texto en la parte inferior de la columna.
- **p**: alinea el texto en la parte superior de la columna.

```
\begin{table}[H]
  \begin{center}
    \caption{Ejemplo de tabla con ancho de columna fijo}
    \label{tab: 5}
    \begin{tabular}{|m{5cm}|m{3cm}|m{1cm}|}
      \hline
      1 & 2 & 3 \\
      4 & 5 & 6 \\
      \hline
    \end{tabular}
  \end{center}
\end{table}
```

Table 5: Ejemplo de tabla con ancho de columna fijo

1	2	3
4	5	6

La Tabla 5 tiene ancho de de columna de 5cm, 3cm y 1cm (izquierda a derecha).

9.4 Combinar filas y columnas

Se requiere del paquete **multirow** para poder combinar columnas y filas.

`\usepackage{multirow}`

Se debe tener cuidado con los parámetros que reciben los comandos `\multirow{ }{ }{ }` y

`\multicolumn{ }{ }{ }` ya que varían un poco: para ambos, el primer parámetro recibe cuántas columnas o filas se van a combinar y el tercero indica el texto o contenido de dichas filas o columnas combinadas, el segundo parámetro varia dependiendo si se combina filas o columnas, para el primer caso, el parámetro recibe el ancho de fila (utilizando unidades de L^AT_EX) y, para el segundo caso, recibe la alineación de las columnas (l, c, r):

```
\begin{table}[H]
  \begin{center}
    \caption{Ejemplo con tabla con columnas combinadas}
    \label{tab: 6}
    \begin{tabular}{c|c|c}
      \hline
      \multicolumn{3}{c}{3 celdas combinadas} \\
      \hline
      1 & 2 & 3 \\
      4 & 5 & 6 \\
      \hline
    \end{tabular}
  \end{center}
\end{table}

\begin{table}[H]
  \begin{center}
    \caption{Ejemplo con tabla con filas combinadas}
    \label{tab: 7}
    \begin{tabular}{m{5cm}|m{3cm}|m{1cm}}
      \hline
      1 & 2 & 3 \\
      \hline
      \multirow{2}{5cm}{2 filas combinadas} & 4 & 5 \\
      & 6 & 7 \\
      \hline
      8 & 9 & 10 \\
      \hline
    \end{tabular}
  \end{center}
\end{table}
```

```
\end{table}
```

Nota: fijarse bien en como se distribuyen los elementos de cada celda si se desea combinar, ya sea filas o columnas, en el ejemplo anterior, por ejemplo, dependiendo si la tabla tendrá tres columnas, *multicolumn* solo puede tomar como máximo tres columnas a combinar, si tiene cinco filas, *multicolumn* se puede poner en cualquiera de esas cinco columnas.

Table 6: Ejemplo con tabla con filas combinadas

3 celdas combinadas		
1	2	3
4	5	6

Table 7: Ejemplo con tabla con columnas combinadas

1	2	3
2 filas combinadas	4	5
	6	7
8	9	10

Repetimos, se debe tener cuidado combinando filas y columnas, para el primer caso, y con un ancho de columna fijo, el ancho de la columna donde se combinarán filas debe coincidir, tanto en la tabla, como en el comando **multirow** (*Tabla 6*), y el número de columnas a combinar debe coincidir con el número real de columnas en una tabla con **multicolumn** (*Tabla 7*).

10 Figuras

Las **figuras** son muy similares a las imágenes en \LaTeX , solo que este requiere de los comandos `\begin` y `\end`, seguido de `{figure}` para indicar una figura, los corchetes de ubicación `[]` también hacen aparición aquí, se requiere el paquete **graphicx**, importar las imágenes a esta plataforma para así adjuntarlas al documento.

Al igual que las imágenes, se adjuntan utilizando el comando `\includegraphics{}`, que de igual forma, tiene el parámetro del tamaño entre corchetes y la figura propiamente dentro de llaves, la cual debe haber sido importada a esta plataforma previamente. La *Figura 4* es el ejemplo de una imagen contenida por una figura:

```
\begin{figure}[H]
  \begin{center}
    \includegraphics [width=10cm]{ejemplo_latex.png}
  \end{center}
\end{figure}
```

Figure 4: Figura de ejemplo



Nota: se recomienda trabajar las imágenes como figuras, ya que este contenedor permite asignarle nombres y etiquetas a las imágenes, como se ve en el ejemplo anterior.

11 Comandos personalizados

El comando `\newcommand` nos permite tomar otros comandos y crear uno nuevo o uno personalizados, de tal forma que nos ahorre algo de tiempo, escritura y código.

11.1 Sin parámetros

Supongamos que queremos un acceso más rápido para el comando `\mathbb` (previamente debimos haber agregado el paquete **amsfonts** para que este comando pudiera servir), para escribir el símbolo de los números reales, utilizamos `newcommand{\mathbb}`, en el primer par de

```
\usepackage{asmfonts}

\newcommand{\R}{\mathbb{R}}

\begin{document}
  \R
\end{document}
```

11.2 Con parámetros

```
\begin{document}
  \newcommand{\M}[2]{
    \begin{bmatrix}
      #1\\
      #2\\
    \end{bmatrix}
  }
\end{document}
```

Segundo intento, le pasamos a `\M{}{}` los valores a y b, el resultado es:

12 Matemáticas

Las matemáticas, fórmulas y ecuaciones son el área fuerte de \LaTeX .

12.1 Fórmulas y área de trabajo de fórmulas

- **Sin espacio dedicado:** una ecuación o fórmula, puesta en el documento de forma sencilla y sin dedicarle un espacio dedicado, requiere de dos símbolos de dolar (\$): \$ecuación\$. Las siguientes dos ecuaciones solamente están puestas con un \$ al inicio y al final, una tras otra, sin comando de salto de línea:

$$e^2, e^{1/2}$$

- **Con espacio dedicado:** dos símbolos de dolar (\$\$) al inicio y final indica el comienzo de un área de trabajo para ecuaciones, la siguiente ecuación están dentro de dicho área y por ello están centradas en el documento:

$$e^{1/3\pi}$$

- **El entorno Equation:** los comandos `\begin{}` y `\end{}`, donde entre sus llaves se escribe **equation**, funcionan para así tener ecuaciones centradas y enumeradas.

$$\begin{equation} \Pi = 3.1416 \end{equation}$$

$$\Pi = 3.1416 \tag{3}$$

- El comando **cdot** pone un punto, en este caso multiplicación, dentro de un texto, fórmula o ecuación.

$$4 \cdot 5$$

- Al principio del documento, podemos usar paquetes dentro del documento para hacerlo más rico y completo, uno de ellos es **amsmath**, te proporciona más comandos matemáticos

$$\usepackage{amsmath}$$

12.2 Alineación de ecuaciones: Align, Slip y Multiline

12.2.1 Align

Si buscamos que nuestras ecuaciones se mantengan alineadas correctamente, centradas y enumeradas, podemos utilizar los comandos `\begin{}` y `\end{}`, donde entre sus llaves se escribe **align**. Esto puede ser usado como cuando se muestra el desarrollo de una ecuación, integral, derivada u otro problema matemático, por ejemplo, el desarrollo de un límite:

```
\begin{align}
\label{ec: ecuacion3}
e &= \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \\
\label{ec: ecuacion4}
&= \lim_{n \rightarrow 0} (1+t)^{\frac{1}{t}}
\end{align}
```

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \quad (4)$$

$$= \lim_{n \rightarrow 0} (1+t)^{\frac{1}{t}} \quad (5)$$

La *Ecuación 4* y *5* muestran un ejemplo de como podríamos tener presente un ejemplo de un desarrollo de ecuaciones sin alinear (símbolos de igualdad dispares) y que, además, al ser un desarrollo de un solo problema, se muestran como ecuaciones independientes. Podemos lograr que este desarrollo se muestre como una sola ecuación.

12.2.2 Slipt

Para hacer que las ecuaciones dentro del desarrollo de una misma estén correctamente alineadas con respecto al símbolo de igualdad, se pone el símbolo `&` antes de todos los símbolos de igualdad de todo el desarrollo para que así, las siguientes ecuaciones se alineen correctamente y, en vez de usar *align* dentro de *begin* y *end*, volvemos a usar *equation*, además de que toda la ecuación estará dentro de otro comando *begin* y *end* donde su contenido ahora será **slipt**.

```
\begin{equation}
\label{ec: 6}
\begin{split}
e &= \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \\
&= \lim_{n \rightarrow 0} (1+t)^{\frac{1}{t}}
\end{split}
\end{equation}
```

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \quad (6)$$

$$= \lim_{n \rightarrow 0} (1+t)^{\frac{1}{t}}$$

La *Ecuación 6* ahora si está bien centrada y enumerada a un solo desarrollo de problema matemático.

12.2.3 Multiline

En caso de tener una ecuación muy larga, se puede usar los comandos `\begin{}` y `\end{}`, donde entre sus llaves se escribe **multiline** para hacer saltos de línea a lo largo de toda la ecuación, para lograr esto, supongamos que tenemos una ecuación matemática larga y la queremos dividir en cuatro partes y ponerlas una debajo de otra, para separar estas partes las encerramos entre `\\` (al inicio y final) y con eso basta, todas las partes deben ser encerradas entre estas barras, tanto la primera (inicial) como la última (final).

La estructura es exactamente la misma al ejemplo anterior, lo único que cambia es precisamente el tipo de *begin* y *end* (con la palabra que ya mencionamos anteriormente) y se usan dos diagonales invertidas en los puntos de la ecuación donde se quiera hacer una separación.

```
\begin{multiline}
  \label{ec: ecuacion6}
  \frac{1}{2}+\frac{2}{2}+\frac{2}{2}+\frac{3}{2}+\frac{4}{2}+\frac{5}{2}+\frac{6}{2}+\frac{7}{2}+\frac{8}{2}+\frac{9}{2}+\frac{10}{2}+\frac{1}{3}+\frac{2}{3}+\frac{3}{3}+\frac{4}{3}+\frac{5}{3}+\frac{6}{3}+\frac{7}{3}+\frac{9}{3}+\frac{10}{3}=0
\end{multiline}
```

$$\frac{1}{2} + \frac{2}{2} + \frac{2}{2} + \frac{3}{2} + \frac{4}{2} + \frac{5}{2} + \frac{6}{2} + \frac{7}{2} + \frac{8}{2} + \frac{9}{2} + \frac{10}{2} + \frac{1}{3} + \frac{2}{3} + \frac{3}{3} + \frac{4}{3} + \frac{5}{3} + \frac{6}{3} + \frac{7}{3} + \frac{9}{3} + \frac{10}{3} = 0 \quad (7)$$

La *Ecuación 7* es muy larga y no tiene espaciado dentro de **multiline**.

```
\begin{multiline}
  \label{ec: ecuacion7}
  \\ \frac{1}{2}+\frac{2}{2}+\frac{2}{2}+\frac{3}{2}+\frac{4}{2} \\ \frac{5}{2}+\frac{6}{2}+\frac{7}{2}+\frac{8}{2}+\frac{9}{2} \\ \frac{10}{2}+\frac{1}{3}+\frac{2}{3}+\frac{3}{3}+\frac{4}{3} \\ \frac{5}{3}+\frac{6}{3}+\frac{7}{3}+\frac{9}{3}+\frac{10}{3}=0 \\
\end{multiline}
```

$$\begin{aligned} & \frac{1}{2} + \frac{2}{2} + \frac{2}{2} + \frac{3}{2} + \frac{4}{2} \\ & + \frac{5}{2} + \frac{6}{2} + \frac{7}{2} + \frac{8}{2} + \frac{9}{2} \\ & + \frac{10}{2} + \frac{1}{3} + \frac{2}{3} + \frac{3}{3} + \frac{4}{3} \\ & + \frac{5}{3} + \frac{6}{3} + \frac{7}{3} + \frac{9}{3} + \frac{10}{3} = 0 \end{aligned} \quad (8)$$

La *Ecuación 8* está dividida en cuatro partes, centrada y con numeración de ecuación, todo gracias a **multiline**.

12.3 Fracciones

12.3.1 Sencillas

- Se utiliza el comando de `\frac{ }{ }` para una fracción, es otra forma de escribirlas: el primer par de llaves es el numerador, el segundo par el denominador.

$$\frac{1}{\frac{3}{n}}$$

$$\frac{2}{5} \\ \left(1 + \frac{3}{n}\right)$$

- La ecuación anterior presenta los paréntesis muy pequeños, se aumentan con el comando `\left` y `\right`, ambos al principio y final de su respectivo paréntesis. En otras palabras, paréntesis que se ajustan automáticamente.

$$\left(\frac{1}{\frac{3}{n}}\right)^n$$

$$\left(1 + \frac{1}{3}\right)^n$$

12.3.2 Fracciones en fracciones

- No es algo más que poner dentro de las llaves de `frac` otro comando `frac`, así como se desee para poner fracciones dentro de fracciones.

$$\frac{1}{\frac{\frac{3}{5}}{1 + \frac{2}{2n+1}}}$$

$$\frac{\frac{1}{\frac{3}{5}}}{1 + \frac{2}{2n+1}}$$

- Si se pone el comando `\ddots` crea tres puntos diagonales que significan que una ecuación continua un camino con patrón visible:

$$\ddots$$

12.4 Límites

Los límites se escriben utilizando el comando `\lim`. Para agregar valores debajo de la palabra *lim*, escribimos el comando `\lim`, seguido usamos el guión bajo (`_`), entre llaves, escribimos la expresión del límite (separados por el comando `\to`). El primero ejemplo a continuación es un límite simple sin valores, el segundo ya tiene un valor escrito debajo de la palabra *lim* (*n* a infinito y *n* a 8):

$$\begin{aligned} & \lim_{n \rightarrow \infty} \left(1 + \frac{1}{3}\right)^n \\ & \lim_{n \rightarrow 8} \left(1 + \frac{1}{3}\right)^n \end{aligned}$$

Nota: el comando `\infty` es un símbolo de infinito en el ambiente de matemáticas.

12.5 Raíces

Para poner una raíz se usa el comando `\sqrt[\text{nivel}]{\text{contenido}}`. Este comando tiene dos parámetros, uno dentro de corchetes y otro dentro de llaves, el primero es para indicar el nivel de raíz (raíz cuadrada/cúbica/cuarta...

), y el otro para el contenido (raíz de 4, 16, 25, x...), pero si se quiere una raíz simple podemos usar solo los parámetros entre llaves. El símbolo de factorial no tiene un comando como tal, solo es el símbolo `!`.

$$\begin{aligned} & \sqrt{25} \\ & \sqrt[3]{4!} \end{aligned}$$

12.6 Sumatorias

Las sumatorias requieren el comando `\sum`, para poner valores debajo de la sumatoria se aplica lo mismo que con los límites, que es poner guión bajo seguido de llaves (`_{\text{valor}}`), para poner valores encima de la sumatoria, debemos poner el símbolo de elevar a tal potencia y, entre llaves, el valor correspondiente (`^{\text{valor}}`, 10, 100,...).

$$\sum_{n=0}^{\infty} \frac{1}{n!}$$

12.7 Integrales

Se usa el comando `\int` para integrales, si es definida, se usa el guión bajo y potencia para indicar los valores definidos de la integral, si se quiere más de una integral en la misma, se ponen todas las i's que se quieran al inicio del comando (`iiint`).

- Definidas

$$\int_1^2 f(x) dx$$

$$\int_1^2 f(x) dx$$

- Indefinidas

$$\int f(x) dx$$

$$\int f(x) dx$$

- Múltiples integrales

$$\iiint_0^2 f(x, y, z) dx dy dz$$

$$\iiint_0^2 f(x, y, z) dx dy dz$$

12.8 Vectores

Se usa el comando `\vec{}`, dentro de sus llaves ponemos el nombre del vector, o la letra que lo caracterizará, si el vector tiene valores, estos valores se indican usando el nombre del vector, seguido de un guión bajo y el número del valor.

- Símbolo

$$\vec{a}$$

$$\vec{a}$$

- Con valores

$$\vec{a} = \langle v_1, v_2, v_3 \rangle$$

$$\vec{v} = \langle v_1, v_2, v_3 \rangle$$

- Con operaciones

$$\vec{a} \cdot \vec{b}$$

$$\vec{a} \cdot \vec{b}$$

12.9 Matrices

Se usa el comando `\begin{}` y `\end{}`, y dentro de sus llaves se usa el comando **bmatrix** para crearla, cada item dentro de una fila se separa con el símbolo `&`, si la matriz tiene tres elementos en la primera fila, se separan con dicho carácter, y para finalizar dicha fila, se ponen las dos diagonales invertidas `\\`(salto de línea), así sucesivamente hasta armar la matriz deseada.

```
$$\begin{bmatrix}
  1 & 2 & 3 & 4 \\
  5 & 6 & 7 & 8 \\
  9 & 10 & 11 & 0 \\
\end{bmatrix}$$
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 0 \end{bmatrix}$$

Referencias

Bazett, T. (2022a, February 15). *How to write a thesis using latex **full tutorial***. <https://youtu.be/zqQM66uAig0>

Bazett, T. (2022b, February 15). *Intro to latex : Learn to write beautiful math equations*. <https://youtu.be/Jp0lPj2-DQA>

Overleaf. (2022, February 15). *Overleaf help*. <https://www.overleaf.com/learn>