

# Conceptos básicos

Francisco Galindo

15 de julio de 2024

## Índice

<b>1. Objetivos de estas sesiones</b>	<b>2</b>
<b>2. ¿Qué son T<sub>E</sub>X y L<sup>A</sup>T<sub>E</sub>X?</b>	<b>2</b>
2.1. ¿Qué gano al aprender L <sup>A</sup> T <sub>E</sub> X?	2
<b>3. Sintaxis básica</b>	<b>3</b>
3.1. Escribiendo texto	4
3.1.1. Espacios en blanco	4
3.1.2. Secciones, subsecciones, etc.	4
3.1.3. <i>Itálicas</i> y <b>negritas</b>	5
3.2. Comandos	5
3.3. Ambientes	5
3.4. Grupos	5
3.5. Comentarios	6
3.6. Clases y paquetes	6
<b>4. Agregando contenido al documento</b>	<b>6</b>
4.1. Escribiendo matemáticas	6
4.1.1. Ambientes para matemáticas	6
4.1.2. Símbolos matemáticos y	7
4.1.3. Sumas, integrales y límites	7
4.1.4. Matrices	8
4.2. Notas	9
4.2.1. Notas al pie	9
4.2.2. Notas al margen	10
4.3. Enlaces	10
4.4. Agregando enlaces en el texto	11
4.4.1. Enlaces web	11
4.4.2. Agregando enlaces arbitrarios	11

<b>5. Dándole formato al documento</b>	<b>12</b>
5.1. Interlineado . . . . .	12
5.2. Espaciado entre párrafos . . . . .	12
5.3. Alineación de texto . . . . .	13
<b>Referencias</b>	<b>13</b>

## 1. Objetivos de estas sesiones

El objetivo de las primeras sesiones de este curso es que ustedes puedan, ya sea por gusto o por necesidad, crear un documento en L<sup>A</sup>T<sub>E</sub>X, agregando la gran mayoría de los elementos a los que ya están acostumbrados en otros sistemas de procesamiento de textos, como la justificación de texto, adición de imágenes y de tablas. Además, conocerán algunos de los elementos que hacen único a L<sup>A</sup>T<sub>E</sub>X con respecto a los sistemas más populales, como su composición de ecuaciones y su manejo de bibliografía.

## 2. ¿Qué son T<sub>E</sub>X y L<sup>A</sup>T<sub>E</sub>X?

T<sub>E</sub>X es un *lenguaje de marcado*<sup>1</sup> creado por Donald Knuth para componer documentos con un aspecto consistente y atractivo. Knuth estaba inconforme con la calidad cada vez menor en la tipografía de su libro *El arte de programar computadoras*. En un año sabático que se tomó (1978) inició el desarrollo del lenguaje, que se consideró *feature-complete*<sup>2</sup> en 1985 y desde entonces las únicas actualizaciones hechas son correcciones de *bugs*.

L<sup>A</sup>T<sub>E</sub>X es un sistema de preparación de documentos [1] creado por Leslie Lamport en 1984. Suele usarse para publicaciones técnicas o científicas, pero puede usarse para preparar casi cualquier tipo de documento. Se trata de un grupo de comandos de T<sub>E</sub>X cuyo propósito es simplificar el uso de T<sub>E</sub>X. El desarrollo de L<sup>A</sup>T<sub>E</sub>X se basa en la idea de que es mejor separar las tareas de diseño de aquellas de escritura.

### 2.1. ¿Qué gano al aprender L<sup>A</sup>T<sub>E</sub>X?

La forma de trabajar en L<sup>A</sup>T<sub>E</sub>X es bastante diferente a comparación de sistemas de procesamiento de textos como *Microsoft Word*, *Google Docs*, o *Libre Office Writer*. En estos últimos, la edición del texto se hace en el mismo lugar en el que se hacen las ediciones de estilo. Es decir, el contenido del documento se edita en el mismo lugar y al mismo tiempo que su formato (fuente, tamaño de letra, justificación, etc.).

L<sup>A</sup>T<sub>E</sub>X, por su parte, trabaja “editando código” de manera similar a lo que se haría al utilizar un lenguaje de programación u otro lenguaje de marcado como HTML. El escritor crea un archivo de texto con todo el contenido del documento y los comandos correspondientes de manera estructurada de acuerdo con su significado e importancia, y mediante un compilador, este archivo de texto (archivo fuente) se convierte a un formato que pueda ser leído por el

<sup>1</sup>[https://en.wikipedia.org/wiki/Markup\\_language](https://en.wikipedia.org/wiki/Markup_language)

<sup>2</sup>[https://en.wikipedia.org/wiki/Software\\_release\\_life\\_cycle#Feature-complete](https://en.wikipedia.org/wiki/Software_release_life_cycle#Feature-complete)

público, (por ejemplo, formato PDF). El diseño estético del documento generalmente se hace por separado, muchas veces por otra persona.

Este cambio de paradigma trae consigo una serie de ventajas y de desventajas. De primeras, se pierden algunas cosas:

- No necesariamente puedes ver la versión final del documento mientras lo editas.
- Tienes que aprender todos los comandos necesarios para escribir tu documento.
- Muchas veces es difícil que el documento se vea *exactamente* como deseas.

Sin embargo, la filosofía de uso de  $\text{\LaTeX}$  brinda muchas ventajas:

- Se puede editar el contenido del documento con cualquier editor de texto. No estás limitado a utilizar cierto editor como es el caso con los programas.
- Suponiendo que tú o alguien más ya definió el estilo del documento, puedes enfocarte enteramente en la creación de la estructura conceptual y contenidos del documento.
- La generación de elementos repetitivos como índices, notas al pie, y referencias se hace de manera automática.
- Debido a que el archivo fuente del documento es texto plano, puede ser compartido, entendido y modificado para las necesidades de otras personas. Es más difícil saber cómo alguien hizo que un documento de *Word* se viera de cierta manera que en  $\text{\LaTeX}$ .
- Elementos como ecuaciones, tablas, y figuras pueden ser generados de manera programática con cualquier otro lenguaje.
- Puede utilizarse cualquier otro lenguaje de marcado para crear un documento, que sea más sencillo de usar, y posteriormente convertirlo a un documento de  $\text{\LaTeX}$  a partir del cuál se pueda generar un PDF o archivo similar.

### 3. Sintaxis básica

Un documento de  $\text{\LaTeX}$  es un archivo en texto plano que puede ser escrito en cualquier editor de texto. Contienen el texto del documento así como los comandos de  $\text{\LaTeX}$  necesarios para componer el texto. Un ejemplo muy básico se ve así:

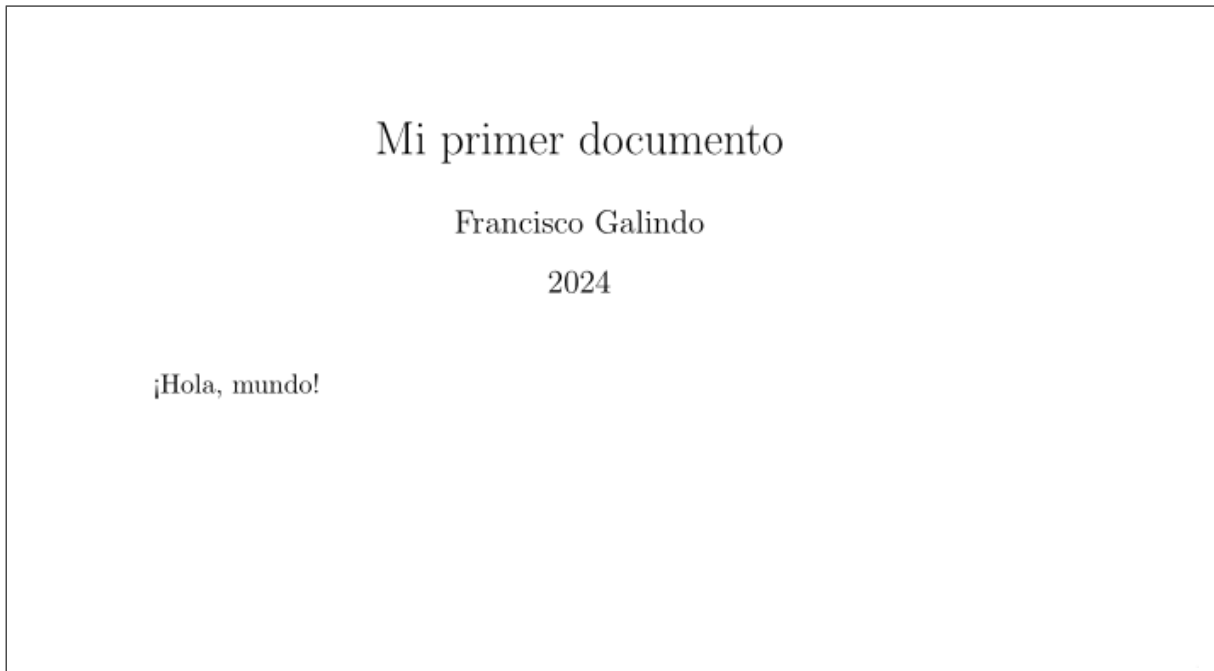
```
\documentclass{article}

\title{Mi primer documento}
\author{Francisco Galindo}
\date{2024}

\begin{document}
\maketitle
¡Hola, mundo!
\end{document}
```

- Este documento es un artículo.
- Su título es “Mi primer documento”.
- Su autor es Francisco Galindo.
- Fue escrito en 2024.
- El documento consiste de un título seguido del texto “¡Hola, mundo!”.

Se obtiene el siguiente resultado:



## 3.1. Escribiendo texto

### 3.1.1. Espacios en blanco

Cualquier grupo de espacios en blanco se verá, en el

### 3.1.2. Secciones, subsecciones, etc.

Para jerarquizar el contenido del documento, existen los comandos `\section`, `\subsection`, `\subsubsection`, `\paragraph`, y `\subparagraph`:

```
\section{Una sección}
```

```
Consectetur laboriosam nulla libero quisquam quos..
```

```
\subsection{Una subsección}
```

```
Consectetur magnam esse illum nobis enim magni?

\subsubsection{Una subsubsección}

Elit eius quas eveniet delectus doloremque.

\paragraph{Un párrafo}: Amet alias doloribus sequi
Natus repellat.

\subparagraph{Subpárrafo}: Adipisicing amet Earum fugit at
qui illum praesentium?
```

### 3.1.3. *Itálicas* y negritas

## 3.2. Comandos

Un comando es un aglomerado de operaciones que  $\text{\LaTeX}$  ejecutará al momento de que el comando sea escrito. La mayoría de los comandos en  $\text{\LaTeX}$  son una palabra con una anti-diagonal ( $\backslash$ ) al principio. Un comando puede llegar a necesitar un argumento, que es escrito entre llaves ( $\{\}$ ) al final del nombre del comando. Algunos comandos pueden aceptar parámetros opcionales, que son añadidos después del nombre del comando y antes del argumento del mismo dentro de corchetes ( $[\ ]$ ) y separados entre comas ( $,$ ). Una llamada a un comando se ve así:

```
\nombredelcomando[parámetro1, parámetro2, ...]{argumento}
```

## 3.3. Ambientes

Un ambiente funciona de manera similar a un comando común, pero generalmente tienen un efecto sobre una parte más larga del documento, como un párrafo entero en lugar de una sola oración. Su sintaxis de uso es:

```
\begin{ambiente}
  Texto afectado por el ambiente...
\end{ambiente}
```

## 3.4. Grupos

Algunos comandos funcionan a manera de *switch*, es decir, que afectan al resto del documento después de haber sido ejecutados:

```
\begin{document}
  Este texto es normal
```

```
\itshape

Este texto está en itálicas.
...
Todo el documento está en itálicas ahora.
...
\end{document}
```

Para limitar el ámbito de influencia (*scope*) de este tipo de comandos, se hace uso de grupos. Un grupo puede crearse al encerrar la parte del documento que se quiere afectar (incluyendo al *switch*) en llaves (`{}`). También pueden utilizarse los comandos `\bgroup` y `\egroup` para iniciar y terminar un grupo, respectivamente.

```
\begin{document}
Texto normal {\itshape texto en itálica} más texto normal.

Texto normal \bgroup \itsahpe texto itálica \egroup{} texto normal.
\end{document}
```

### 3.5. Comentarios

El símbolo `%` es utilizado para denotar un comentario. Cuando  $\text{\LaTeX}$  está procesando un documento y se encuentra con un `%`, ignora el resto de la línea. De esta manera, un comentario se puede utilizar para escribir notas en el archivo fuente que no serán visibles en la versión final del documento.

### 3.6. Clases y paquetes

Una clase define cómo se ve, a groso modo el documento que estás escribiendo. Se trata de un grupo de comandos que se ejecutan para darle el estilo deseado al documento. Se seleccionan con el comando `\documentclass`. Se hablará más de clases de documentos en próximas sesiones.

Un paquete es similar a una biblioteca en un lenguaje de programación. Incluyen una serie de características adicionales que puedes utilizar en tu documento, como el uso de colores, adición de imágenes o inclusión de nuevos comandos y ambientes.

## 4. Agregando contenido al documento

### 4.1. Escribiendo matemáticas

#### 4.1.1. Ambientes para matemáticas

Al momento de escribir expresiones matemáticas,  $\text{\LaTeX}$  necesita saber dónde empiezan y terminan las mismas. Existen dos ambientes distintos para presentar texto matemático:

- *inline*: las fórmulas aparecen como parte de un párrafo de texto. Un ejemplo es este:  
 $(a - b)(a + b) = a^2 - b^2$ .
- *display*: se utiliza para escribir expresiones que no son parte de un párrafo, es decir, que aparecen en una línea por sí mismas. Por ejemplo:

$$(a - b)(a + b) = a^2 - b^2.$$

Los ambientes correspondientes a ambos tipos de expresiones se pueden utilizar así:

```
\begin{math} % modo inline
...
\end{math}

\begin{displaymath} % modo display
...
\end{displaymath}
```

Como un documento puede contener muchas (¡muchas!) ecuaciones,  $\text{\LaTeX}$  tiene “abreviaciones” para entrar en estos modos,  $\text{\textbackslash}(\dots\text{\textbackslash})$  para el modo *inline* y  $\text{\textbackslash}[\dots\text{\textbackslash}]$  para *display*. Por ejemplo, este pedazo de código tiene el siguiente resultado:

```
Una ecuación en modo \textit{inline} en medio de una oración:
\(\text{x} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\text{).}
```

```
Esta es la misma ecuación en modo \textit{display}:
\[\text{x} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\text{.}\]
```

Una ecuación en modo *inline* en medio de una oración:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ .  
Esta es la misma ecuación en modo *display*:

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

#### 4.1.2. Símbolos matemáticos y

#### 4.1.3. Sumas, integrales y límites

El símbolo de una integral puede escribirse con el siguiente comando:  $\text{\textbackslash}int_{\text{\textit{inf}}}{\text{\textit{sup}}}$ , donde *inf* y *sup* son los límites de integración. Pueden escribirse integrales múltiples, así como integrales cerradas como esta:

$$\oint_{\gamma} F \cdot dr = \iint_S \nabla \times F \cdot dS.$$

Código	Res. <i>inline</i>	Res. <i>display</i>
<code>\int f(x)\,dx</code>	$\int f(x) dx$	$\int f(x) dx$
<code>\int_{\{a\}^{\{b\}} f(x)\,dx</code>	$\int_a^b f(x) dx$	$\int_a^b f(x) dx$
<code>\iint_{\{S\}} f(x,y)\,dA</code>	$\iint_S f(x,y) dA$	$\iint_S f(x,y) dA$
<code>\iiint_E f(x,y,z)\,dV</code>	$\iiint_E f(x,y,z) dV$	$\iiint_E f(x,y,z) dV$
<code>\oint_{\{\gamma\}} f(z)\,dz</code>	$\oint_{\gamma} f(z) dz$	$\oint_{\gamma} f(z) dz$
<code>\idotsint f(x_1, \dots, x_k)</code>	$\int \cdots \int f(x_1, \dots, x_k)$	$\int \cdots \int f(x_1, \dots, x_k)$

Tabla 1: Ejemplos de varios tipos de integrales con ayuda del paquete

#### 4.1.4. Matrices

$$\iint_V \mu(u, v) du dv$$

**Teorema 1.** El valor de  $\int_{-\infty}^{\infty} e^{-x^2} dx$  es

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}.$$

*Demostración.*

$$\begin{aligned}
\left( \int_{-\infty}^{\infty} e^{-x^2} dx \right)^2 &= \left( \int_{-\infty}^{\infty} e^{-x^2} dx \right) \left( \int_{-\infty}^{\infty} e^{-y^2} dy \right) && \text{por (2)} \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^2} e^{-y^2} dx dy \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} dx dy \\
&= \int_0^{2\pi} \int_0^{\infty} e^{-r^2} r dr d\theta \\
&= \int_0^{2\pi} \left[ -\frac{e^{-r^2}}{2} \right]_{r=0}^{r=\infty} d\theta \\
&= \int_0^{2\pi} \frac{1}{2} d\theta \\
&= \pi
\end{aligned}$$

□



$$e^{i\pi} + 1 = 0 \quad (1)$$

El resultado obtenido en la ecuación 1 es de gran importancia en el análisis complejo.

Considere el operador lineal  $A : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , definido por una matriz  $\mathbf{A}$  de la siguiente manera:

$$\mathbf{A}v = w$$

que también puede escribirse como

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}.$$

Si se cumple que

$$\mathbf{A}v = \lambda v \quad (2)$$

entonces se dice que  $v$  y  $\lambda$  son un *eigenvector* y un *eigenvalor* de  $A$ , respectivamente.

La ecuación 2 puede escribirse como

$$(\mathbf{A} - \lambda I)v = 0,$$

y, resolviendo para  $\lambda$ , se sabe que esta ecuación sólo tiene una solución no trivial cuando

$$\det(\mathbf{A} - \lambda I) = 0,$$

que también se puede escribir como

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{vmatrix} = 0.$$

## 4.2. Notas

### 4.2.1. Notas al pie

Una nota al pie de página puede crearse con el comando `\footnote`. Estas notas son enumeradas de manera automática:

Aquí tengo un párrafo en el que quiero hacer una aclaración\footnote{Soy una nota al pie.}, aunque no me gustaría interrumpir el texto. Puedo crear varias notas al pie\footnote{Soy otra nota.}.

Aquí tengo un párrafo en el que quiero hacer una aclaración<sup>a</sup>, aunque no me gustaría interrumpir el texto. Puedo crear varias notas al pie<sup>b</sup>.

---

<sup>a</sup>Soy una nota al pie.

<sup>b</sup>Soy otra nota.

Si no es conveniente indicar que existe la nota y escribir el contenido de la nota al mismo tiempo, pueden utilizarse los comandos `\footnotemark` y `\footnotetext` para hacer ambas cosas en pasos separados. El siguiente pedazo de código tiene el mismo resultado que haber utilizado el comando `\footnote` directamente.

```
Sé que va a haber una nota aquí\footnotemark{}. Aut animi
quasi aliquam cumque magnam autem Impedit deserunt quia
nulla voluptates similique similique consectetur,
dignissimos! Provident ab?
```

```
\footnotetext{Este es el texto de la nota.}
```

#### 4.2.2. Notas al margen

El comando `\marginpar{texto}` creará un párrafo en el margen de la página a la altura en la que haya sido llamado. Por defecto, estos párrafos se colocan “afuera” de la página:

```
Ipsum corporis ea doloribus corrupti ullam sequi Suscipit
omnis dolores doloremque magni optio! Et dolor obcaecati
\marginpar{Soy una nota al margen} nesciunt ut repellat
Provident deleniti hic illo quam iusto maxime minima harum?
Cum dolores? Elit corporis recusandae atque minima voluptatum
necessitatibus Cum neque quaerat suscipit non sed! Quasi
sapiente ratione incidunt explicabo id beatae Aliquid beatae
corporis sapiente assumenda culpa Soluta illo nobis
exercitationem.
```

<pre>Ipsum corporis ea doloribus corrupti ullam sequi Suscipit omnis dolores do- loremque magni optio! Et dolor obcaecati nesciunt ut repellat Provident deleniti hic illo quam iusto maxime minima harum? Cum dolores? Elit corporis recusan- dae atque minima voluptatum necessitatibus Cum neque quaerat suscipit non sed! Quasi sapiente ratione incidunt explicabo id beatae Aliquid beatae corporis sapiente assumenda culpa Soluta illo nobis exercitationem.</pre>	<p>Soy una nota al margen.</p>
--	------------------------------------

#### 4.3. Enlaces

Es deseable que, al momento de compilar un documento a algún formato de visualización, como PDF, existan herramientas de navegación para facilitar la vida del lector. El paquete `hyperref` se encarga de gestionar la creación de enlaces dentro del documento. Por ejemplo,

simplemente al importar<sup>3</sup> el paquete, todos los elementos del documento a los que se les haga referencia durante el documento, como figuras, ecuaciones, o los elementos del índice, se convertirán en hipervínculos.

Se puede configurar el paquete para modificar su comportamiento con el comando `\hypersetup`. Por ejemplo, este documento utiliza el comando de la siguiente manera:

```
\hypersetup{
  colorlinks=true, % Para que los enlaces tengan color

  pdftitle={Conceptos básicos de LaTeX}, % Título que
  % aparece en
  % lectores PDF
}
```

Los parámetros modificables pueden encontrarse en este [enlace](#).

## 4.4. Agregando enlaces en el texto

### 4.4.1. Enlaces web

Para crear un enlace para, por ejemplo, una página web puede utilizarse el comando `\url` o `\href`. El primero mostrará el enlace en sí, mientras que el segundo lo esconde para mostrar en su lugar cierto texto:

```
Puedo escribir un enlace directamente:
\url{www.https://wikipedia.org/}.
También puedo hacer que en su lugar aparezca una
\href{https://wikipedia.org/}{oración}.
```

Puedo escribir un enlace directamente: [www.https://wikipedia.org/](https://wikipedia.org/). También puedo hacer que en su lugar aparezca una [oración](#).

### 4.4.2. Agregando enlaces arbitrarios

Es posible crear enlaces que lleven a la parte del documento que se desee. Para esto, se utilizan los comandos `\hyperlink{etiqueta}{texto}` y `\hypertarget{etiqueta}{texto}`:

```
En otra parte del documento, hay una
\hyperlink{importante}{oración importante}.

Esta es una oración
\hypertarget{importante}{de gran importancia}.
```

En otra parte del documento, hay una [oración importante](#).  
Esta es una oración de gran importancia.

---

<sup>3</sup>Es recomendable que este sea el último paquete en importarse, para evitar conflictos.

Opción	Descripción
<code>skip</code>	Indica, de manera explícita, la distancia vertical entre un párrafo y otro.
<code>indent</code>	Configura la longitud de la sangría del primer renglón del párrafo.

Tabla 2: Algunas de las opciones del paquete `parskip`.

## 5. Dándole formato al documento

### 5.1. Interlineado

El paquete `setspace` incluye los comandos necesarios para establecer el interlineado de los párrafos, ya sea en el documento entero o en alguna sección:

- `\doublespacing` para espaciado doble.
- `\onehalfspacing` para espaciado 1.5.
- `\singlespacing` para espaciado simple.

Estos comandos pueden usarse en el preámbulo del documento para que afecten todo el texto, o en alguna zona local por medio de un **grupo**.

### 5.2. Espaciado entre párrafos

Muchas veces, es necesario implementar disposiciones de párrafos donde los mismos están separados por cierto espacio vertical además (o en lugar) de tener una sangría. El paquete `parskip` facilita hacer estas configuraciones de texto. El paquete sólo necesita ser cargado con `\usepackage` para tomar efecto. Dependiendo de las opciones (como las te la tabla 5.2) que sean enviadas al paquete, se puede modificar el espaciado entre párrafos así como el tamaño de la sangría.

Un ejemplo, donde se necesita que no haya sangría, que el espacio entre párrafos sea de `1em` y con interlineado doble sería el siguiente:

```
\documentclass{article}

% ...
\usepackage[indent=0cm, skip=1em]{parskip}
\usepackage{setspace}
\doublespacing
% ...
```

```
\begin{document}
  Blah, blah, blah...
\end{document}
```

### 5.3. Alineación de texto

Por defecto,  $\text{\LaTeX}$  justifica el texto de cada párrafo para que se vea “parejo”, pero a veces es deseable alinear el texto de otras formas. Ya vienen incluidos varios comandos y ambiente para alinear el texto del documentos, pero el paquete `ragged2e` refina estos comandos para que la separación de palabras sea más natural.

El paquete `ragged2e` incluye los siguientes comandos que funcionan a manera de switch:

- `\RaggedRight` para alinear a la izquierda (el lado derecho queda “accidentado”).
- `\RaggedLeft` para alinear a la derecha.
- `\Centering` para centrar texto.
- `\justifying` para justificar texto.

También incluye algunos ambientes con fines similares, pero con la ventaja de la creación implícita de grupos:

- `FlushLeft` para alinear a la izquierda.
- `FlushRight` para alinear a la derecha.
- `Center` para centrar texto.
- `justify` para justificar texto.

## Referencias

- [1] *Introduction to LaTeX*, en. dirección: <https://www.latex-project.org/about/>.