

El presente material fue desarrollado por:

Daniel Lara

Facultad de Ciencias, Escuela Politécnica Nacional

Andrés Merino

Facultad de Ciencias Exactas y Naturales, Pontificia Universidad Católica del Ecuador

La versión actual del material es 1.2-(Mayo 2021). En caso de encontrar inconsistencias o errores en el presente material se pueden comunicar a daniel.lara@alephsub0.org. Para más información puedes visitar nuestro sitio web: alephsub0.org. Si deseas colaborar con el desarrollo de este material, el código fuente está disponible en: https://github.com/alephsub0/LaTeX_Guias.git. Cualquier aporte (*Pull request*) será de gran ayuda para mejorar este material.

© ⓘ Ⓞ Esta obra se encuentra bajo licencia Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) Para más información puede visitar: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Otros colaboradores: Katheryn Yánes

ÍNDICE

1	La filosofía de L^AT_EX	2
2	¿Qué necesitamos?	3
2.1	¿Todo esto para dos palabras?	4
3	Antes de iniciar	5
3.1	Consideraciones sobre el editor	5
3.2	Consideraciones sobre la redacción	5
3.3	Algunos sitios de utilidad	5
4	Generalidades	6
5	Paquetes	7
5.1	El paquete babel	8
5.2	El paquete geometry	9

1. LA FILOSOFÍA DE L^AT_EX

El proceso para la generación profesional de un documento escrito se lo puede dividir en tres etapas:

- Escritura: es cuando el *autor* plasma sus ideas concentrándose únicamente en el fondo del documento más no en su forma (formato), entregando un manuscrito.
- Maquetación: es cuando el *maquetador* decide el formato del documento (tamaño de página, márgenes, número de columnas, tipografía, espaciado, etc.), entregando una lista de instrucciones a seguir.
- Composición: es cuando el *compositor* usa el manuscrito del autor y la lista de instrucciones del maquetaador para generar el documento final a ser publicado.

Con esto en mente, el autor debe *concentrarse únicamente en escribir*, pues es el maquetaador, quien con su experticia en la diagramación de textos, decide el formato adecuado.

Dado que, no siempre contamos con el equipo necesario (maquetadores y compositores) para la generación de un documento escrito profesional, varios gestores de texto (como *Microsoft Word*) generaron entornos para cargar todo este trabajo al autor, proporcionando un sistema que lo que se ve en pantalla es lo que se obtiene como resultado. Esto coloca una carga extra sobre los autores que ahora deben lidiar con el formato del documento en lugar de *enfocarse en escribir*.

Para salvarnos de esto, Donald Knuth y Leslie Lamport (ambos matemáticos) nos regalaron, respectivamente, las siguientes herramientas: T_EX y L^AT_EX. Antes de ver qué son estas herramientas, podemos decir, a muy breves rasgos, qué hacen:

- L^AT_EX es un maquetaador de documentos, y
- T_EX es un compositor de documentos.

Así, si nos ayudamos de estas herramientas, como autores, podemos dedicarnos a nuestra labor, es decir *dedicarnos solo a escribir*.

Por otro lado, siempre es necesario tener cierta comunicación con nuestro maquetaador, por ejemplo, decirle qué partes de nuestro texto son títulos o que partes son una lista numerada; esto lo podemos realizar colocando *marcas* dentro de nuestro texto y dado que no queremos perder atención mientras vamos escribiendo, lo mejor sería que estas marcas también estén de forma escrita, por ejemplo:

```
(esto es un título de sección) Filosofía de LaTeX (aquí termina el título)
</>
El proceso para la generación...
```

Ahora, dado que nuestro maquetaador será un sistema informático, será necesario que estas marcas sean estandarizadas de tal forma que L^AT_EX siempre las entienda, por esta razón, L^AT_EX cuenta con toda una serie de *marcas espaciales*, que se denominan comandos, haciendo que el ejemplo anterior quede de la siguiente manera:

```
\section{Filosofía de LaTeX}
</>
El proceso para la generación...
```

Al conjunto de comando se lo conoce como *lenguaje* \LaTeX o *código* \LaTeX , por lo tanto, podemos decir que \LaTeX es un *lenguaje de marcado*.

Una vez que hemos terminado de escribir nuestro texto, colocando algunas marcas para darle indicaciones a \LaTeX , es necesario realizar un proceso para que nuestro maquetador interprete nuestras marcas y, junto con nuestro compositor, genere el documento final. A este proceso se lo denomina *compilación* es realizado por un *compilador*. El principal compilador del lenguaje \LaTeX se llama, de igual forma, compilador \LaTeX , aunque existen otro como pdf \LaTeX . Al realizar este proceso con el código del ejemplo anterior, obtenemos nuestro resultado final: el inicio de esta sección.

Como acabamos de ver, al referirnos a \LaTeX , podemos estar haciendo mención a varias cosas, pero todas ellas encaminadas a un mismo propósito: *que el autor se concentre solo en escribir su documento*, de todo lo demás se encarga \LaTeX .

2. ¿QUÉ NECESITAMOS?

En primer lugar, dado que solo debemos concentrarnos en escribir nuestro documento y colocar las marcas adecuadas para que \LaTeX haga todo lo demás, necesitamos únicamente un editor de texto plano, de este tipo hay muchos: Bloc de notas, TextEdit, Gedit; sin embargo, dado que vamos a escribir código, es recomendable utilizar un editor especializado para \LaTeX o que tenga complementos para código \LaTeX . Algunas opciones son: \TeX Maker, Kile, WinEdt o Visual Studio Code; pero como se mencionó, basta con un simple Bloc de notas.

Por otro lado, necesitamos todos los programas necesarios para generar la compilación, aunque podríamos instalarlos uno por uno, es mejor instalar todo en conjunto. Para esto, existen las *distribuciones de \LaTeX* que nos instalan todos los programas necesarios¹; algunas de estas distribuciones son: MiK \TeX , Mac \TeX , \TeX Live.

Una vez que tengamos esto, podemos escribir en las siguientes líneas las siguientes líneas en un archivo y llamarlo, por ejemplo, MiPrimerArchivo.tex:

```
\documentclass[11pt]{article}

</> \begin{document}
    Hola Mundo
\end{document}
```

Notemos que la *extensión* del archivo es *.tex*, con esto indicaremos a \LaTeX que este es un archivo donde está el manuscrito que realizamos, es decir, todo archivo que quiera compilarlo con \LaTeX debe tener extensión *.tex*.

Ahora, para pedir a \LaTeX que realice todo el trabajo para generar nuestro documento final, debemos realizar la compilación del archivo MiPrimerArchivo.tex. Esto lo podemos lograr abriendo un *terminal* de nuestro computador en la misma carpeta que tenemos nuestro archivo² y escribir:

```
</> pdflatex MiPrimerArchivo.tex
```

¹Algunas distribuciones vienen también con editores de texto especializados para \LaTeX .

²En Windows, esto se consigue abriendo el explorador de archivos en la carpeta que tengamos nuestro archivo y escribiendo en la barra de direcciones el texto: cmd y dando un enter

Así, pedimos que se compile nuestro archivo utilizando el compilador pdfL^AT_EX. Podemos ver esto en la Figura 1, ejecutamos este comando (damos enter) y L^AT_EX se pondrá manos a la obra y nos devolverá nuestro primer archivo PDF generado utilizando L^AT_EX.

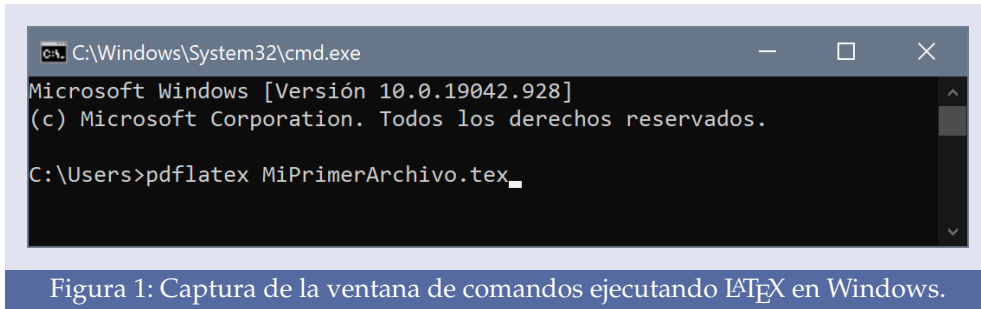


Figura 1: Captura de la ventana de comandos ejecutando L^AT_EX en Windows.

2.1 ¿Todo esto para dos palabras?

Sí, pero con dos indicaciones: primero, los editores especializados tiene «botones» especializados para que hagan todo el proceso de compilación con un solo clic; segundo, es el mismo proceso tanto para nuestro archivo de dos palabras como para un libro de cientos de páginas. Además, ya conocemos brevemente la secuencia de trabajo con L^AT_EX:

1. Escribir nuestro documento.
2. Compilar nuestro documento.

Una consideración adicional, si deseamos obviar todo esto de la instalación, podemos usar servicios en la nube los cuales ya tienen instalado las distribuciones necesarias y nos presentan un editor de texto especializado para escribir código L^AT_EX, la más popular de todas (y en la cual basaremos estas guías) es *Overleaf*, en la cual solo necesitamos crearnos una cuenta y ya podremos iniciar a escribir nuestros documentos. Aquí, podemos dividir nuestro trabajo en *proyectos* e ingresando a cada proyecto ya podremos escribir nuestro texto y compilarlo; podemos ver la interfaz de Overleaf en la Figura 2.

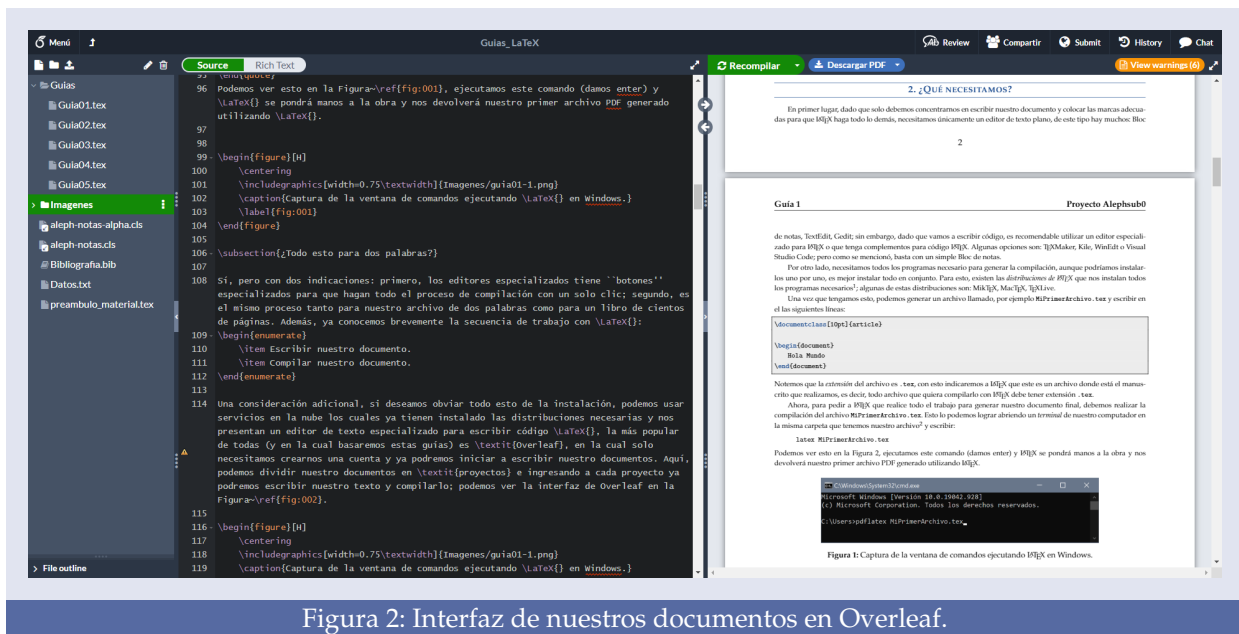
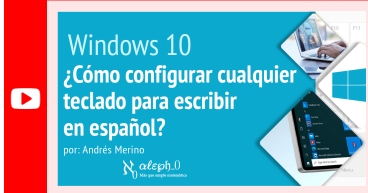


Figura 2: Interfaz de nuestros documentos en Overleaf.

3. ANTES DE INICIAR

3.1 Consideraciones sobre el editor

Al trabajar con \LaTeX , casi la totalidad del tiempo estaremos escribiendo código. Por esta razón para que nuestro trabajo sea eficiente es necesario tener adecuadamente configurado nuestro teclado, un video sobre este tema se puede encontrar en el siguiente enlace:

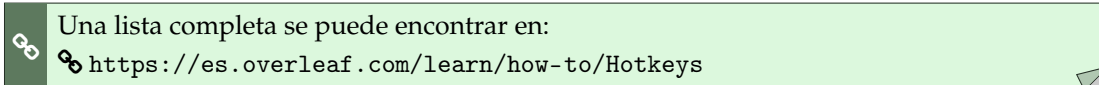


¿Cómo configurar cualquier teclado para escribir en español?
<https://link.alephsub0.org/ConfiguracionTeclado>

Te enseñamos cómo configurar de manera correcta cualquier teclado, esté en español o inglés, para utilizarlo al escribir en español, además, te indicamos cómo usarlo luego de configurado.

Por otro lado, podemos aprovechar las ventajas de los atajos de teclado de nuestro editor y de esta forma realizar un uso casi exclusivo del teclado para el desarrollo de nuestro trabajo. Los principales atajos de teclado del editor de Overleaf son:

- Ctrl + enter para compilar.
- Ctrl + / para comentar el código.
- Ctrl + Home mueve al inicio del documento.
- Ctrl + End mueve al final del documento.
- Ctrl + L mueve a la línea especificada.
- Ctrl + D elimina la línea actual.
- Ctrl + B texto en negrita.
- Ctrl + I texto en itálica.
- Ctrl + F para buscar y reemplazar.
- Shift + Tab para eliminar sangría.



Una lista completa se puede encontrar en:
<https://es.overleaf.com/learn/how-to/Hotkeys>

3.2 Consideraciones sobre la redacción

Al momento de redactar un texto, debemos tener en cuenta la siguiente premisa: Todo dentro de un texto o es un párrafo, o es parte de un párrafo, o es un título, o es un flotante (imagen, tabla, etc.). Dado que, en general, lo que predomina en un texto son párrafos, asumiremos que, a menos que se indique lo contrario, todo lo que escribimos es un párrafo o parte de uno.

3.3 Algunos sitios de utilidad

Antes de comenzar el trabajo en \LaTeX es importante tomar en cuenta los siguientes enlaces para los recursos que nos serán de utilidad a lo largo del curso:

- Página de Overleaf: <https://www.overleaf.com/>
- Página oficial de \TeX Maker: <https://www.xm1math.net/texmaker/>
- Página oficial de Mi \TeX : <http://miktex.org/>
- Red de completa de archivos \TeX : <http://www.ctan.org/>
- Catálogo en línea de paquetes de \LaTeX : http://www.ctan.org/pkg/_A

- Grupo de usuarios T_EX: <http://www.tug.org/>
- Comunidad de usuarios L^AT_EX de Stack Exchange: <https://tex.stackexchange.com/>

4. GENERALIDADES

Regresemos a nuestro archivo `MiPrimerArchivo.tex`:

```
</> \documentclass[11pt]{article}
</> \begin{document}
      Hola Mundo
</> \end{document}
```

En la primera línea de este archivo, podemos notar el comando `\documentclass[11pt]{article}`; en general esta es la sintaxis básica de todo comando de L^AT_EX:

- Todo comando de L^AT_EX inicia con la barra inversa «`\`».
- Luego de la barra inversa, sin espacio, va el nombre del comando (en este caso, el nombre del comando es `documentclass`).
- A continuación, entre corchetes, se coloca los argumentos opcionales; como su nombre lo indica, estos argumentos podría no ser necesarios, en cuyo caso, simplemente se elimina esta parte quedando: `\documentclass{article}`.
- Finalmente, entre llaves, se coloca los argumentos obligatorios; existen comandos que no requieren ningún argumento obligatorio, en esos casos, el comando solo cuenta con la barra inversa y el nombre del comando.

En nuestro caso, nuestro comando `\documentclass[11pt]{article}` sirve para definir el formato general del documento, algunas posibilidades disponibles para el argumento obligatorio son

- `article`;
- `book`;
- `beamer`;
- `standalone`;
- `report`;
- `slides`;
- `letter`;
- `minimal`.

Cada una de estas definirá el formato general del documento como la geometría de la página, el formato para encabezados, títulos, etc. Por otra parte, el argumento opcional `10pt` define el tamaño de fundición (tamaño de letra), otras posibilidades para este argumento son:

- `10pt`;
- `11pt`;
- `12pt`.

Este argumento es opcional dado que, si no lo colocamos, tomará el valor por defecto que, en el caso de `article`, es de `10pt`, es decir, colocar

```
</> \documentclass[10pt]{article}
```

es equivalente a colocar

```
</> \documentclass{article}
```

El siguiente comando que observamos en nuestro archivo es `\begin{document}`, este marca el inicio del **cuerpo del documento**, todo lo que se coloque antes de este comando, se lo denomina **preámbulo del documento**. Finalmente, tenemos el comando `\end{document}`, el cual marca el final del cuerpo del documento.

Todo archivo debe tener, como mínimo, estos tres comandos; el primer comando que aparezca en el documento siempre debe ser `\documentclass` y todo lo que esté luego del comando `\end{document}` será ignorado en la compilación.

Para compilar un documento exitosamente se requiere que siempre exista contenido entre `\begin{document}` y `\end{document}`. Si intentamos compilar un documento con esta sección vacía, este proceso se detendrá sin resultado alguno.

El comando `\begin` tiene un carácter especial pues define lo que se conoce como **ambientes**, por lo cual, siempre que se coloque un comando `\begin`, debe estar emparejado con un comando `\end` para que se delimite el alcance del ambiente. Por ejemplo, para utilizar un ambiente ficticio llamado `foo`, se debe utilizar el siguiente esquema:

```

Texto fuera del ambiente foo.
\begin{foo}
  Texto dentro del ambiente foo.
\end{foo}
Texto fuera del ambiente foo.

```

En los párrafos anteriores, utilizamos el ambiente `document` para escribir todo el contenido de nuestro documento.

5. PAQUETES

\LaTeX cuenta con una abundante cantidad de comandos y ambientes que podemos utilizar para escribir nuestro documento, por ejemplo, el comando `\today`, el cual imprime la fecha en la que se compila el documento. Veamos el siguiente ejemplo, el código:

```

\documentclass[11pt]{article}

\begin{document}
  La fecha en que se escribió este documento es \today.
\end{document}


```

genera el siguiente texto:

La fecha en que se escribió este documento es November 23, 2021.

Como podemos apreciar, se imprimió lo que esperábamos, pero en inglés, ya que es el idioma por defecto en el que trabaja \LaTeX y no le indicamos que íbamos a escribir en español. Para esto, necesitamos introducir lo que se conoce como **paquetes**.

Un **paquete de \LaTeX** es un archivo que incluye comandos destinados a agregar nuevas funciones o redefinir funciones ya existente o facilitar su uso.

La lista oficial de paquetes disponibles, junto con su documentación, se pueden encontrar en:
 <https://ctan.org/https://ctan.org/>

Para incluir un paquete se debe utilizar la siguiente sintaxis, siempre en el preámbulo del documento:

```
</> \usepackage[Opciones del paquete]{Nombre del paquete}
```

Para escribir textos en español, los paquetes básicos que se debe utilizar son los siguientes:

- `babel`: Gestiona el idioma del documento, en nuestro caso, utilizaremos la opción `spanish`.
- `inputenc`: Gestiona la codificación de caracteres³ de entrada, esto nos permitirá escribir directamente desde nuestro teclado los caracteres especiales del español (por ejemplo las tildes). La opción para el paquete depende del editor que estemos utilizando, en nuestro caso, al utilizar Overleaf, colocaremos la opción `utf8`.
- `fontenc`: Gestiona la codificación de caracteres de salida, esto nos permitirá que, al momento de tener nuestro documento PDF y copiar texto de él, los caracteres especiales sean reconocidos correctamente. La opción que utilizaremos es `T1`.

Con los paquetes mencionados es posible comenzar la composición de un documento, quedando nuestro ejemplo de la siguiente manera:

```
\documentclass[10pt]{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
</> \usepackage[spanish]{babel}

\begin{document}
  La fecha en que se escribió este documento es \today.
\end{document}
```

genera el siguiente texto:

 La fecha en que se escribió este documento es 23 de noviembre de 2021.

Conforme avancemos en los contenidos, introduciremos nuevos paquetes que ampliarán las herramientas que disponemos. No obstante, para el desarrollo de un documento, se recomienda trabajar únicamente con los paquetes requeridos para evitar conflictos en la compilación del documento.

5.1 El paquete `babel`

Como mencionamos anteriormente, el paquete `babel` gestiona el idioma del documento, con esto, nos referimos a que adapta la estructura del documento a las particularidades propias del idioma. Algunas de estas particularidades son:

- Nombres de los elementos del documento: por ejemplo, los títulos de resúmenes, tablas de contenidos o la misma fecha que vimos anteriormente, entre otros.

³Ver más: https://es.wikipedia.org/wiki/Codificacion_de_caracteres

- Separación en sílabas de las palabras para cortes al final de cada línea.
- Formato de los párrafos, formato de números romanos, formato de separador decimal, formato de listas, formato de sangría.

El paquete `babel` admite alrededor de 50 idiomas diferentes y dentro de cada idioma tiene varias opciones para configurar detalles especiales. Por ejemplo, para el idioma español el experto en ortotipografía⁴ Javier Bezos planteó y programó las opciones que pueden ser utilizadas por el paquete `babel`.

La lista de opciones para el idioma español del paquete `babel` puede ser consultada en:
<http://www.texnia.com/spanishopt.html>
 Las bases para estas opciones pueden ser consultadas en:
<http://www.texnia.com/spanish2.html>

5.2 El paquete `geometry`

Otro paquete importante, pero no fundamental, es el paquete `geometry`, el cual nos ayuda con una de las primeras tareas a realizar en la preparación de un documento: definir su tamaño, márgenes y otras medidas.

Por ejemplo, si deseamos cambiar el tamaño de hoja a un tamaño A5, en orientación horizontal y con 2 cm de margen a cada lado, basta colocar las siguientes opciones al incluir el paquete:

```
</> \usepackage[a5paper, landscape, margin=2cm]{geometry}
```

En cuanto a tamaño de hoja, existen varios argumentos que pueden ser utilizados, los más comunes pueden ser los de la serie A de la ISO para utilizarlos, estos están descritos en el Cuadro 1. También se puede definir el tamaño de hoja de forma personalizada mediante los argumentos `paperwidth`, `paperheight` o `papersize` de la siguiente manera:

```
</> \usepackage[papersize = {10cm, 20cm}]{geometry}
```

o, de forma equivalente:

```
</> \usepackage[paperwidth = 10cm , paperheight = 20cm]{geometry}
```

Cuadro 1: Algunos tamaños de hojas disponibles		
Argumento	Normativa	Dimensiones
a0paper	ISO - DIN Serie A0	841mm × 1189mm
a1paper	ISO - DIN Serie A1	594mm × 841mm
a2paper	ISO - DIN Serie A2	420mm × 594mm
a3paper	ISO - DIN Serie A3	297mm × 420mm
a4paper	ISO - DIN Serie A4	210mm × 297mm
a5paper	ISO - DIN Serie A5	148mm × 210mm

En lo que respecta a orientación de la hoja, para horizontal se utiliza el argumento `landscape` y para vertical, `portrait`; sin embargo, esta última es la opción por defecto, por lo tanto, podría

⁴Conjunto de normas para el uso correcto de la tipografía.

omitirse. Finalmente, para los márgenes, existen varias formas de definirlos; si queremos tener las mismas medidas en todos los márgenes podemos colocar

```
</> \usepackage[margin = 4cm]{geometry}
```

si queremos dar una misma medida a los márgenes verticales (arriba y abajo) y hacer lo propio con los horizontales (izquierda y derecha), podemos colocar

```
</> \usepackage[vmargin = 2cm, hmargin = 3cm]{geometry}
```

por último, si queremos dar medidas diferentes a cada margen, colocamos

```
</> \usepackage[right = 2cm, left = 1cm, top = 3cm, bottom = 4cm]{geometry}
```

o, de manera equivalente⁵,

```
</> \usepackage[rmargin = 2cm, lmargin = 1cm,
           tmargin = 3cm, bmargin = 4cm]{geometry}
```

El paquete `geometry` dispone de muchas más opciones para controlar varios otros aspectos de la geometría de la hoja y se puede consultar en su documentación.

La documentación del paquete `geometry` puede ser consultada aquí:
<https://ctan.org/pkg/geometry>

Con lo aprendido hasta ahora, tenemos que la estructura base de cualquiera de nuestros documentos puede verse algo así:

```
</> \documentclass[11pt]{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[spanish]{babel}

\usepackage[a4paper, margin = 3cm]{geometry}

\begin{document}
  Mi primer documento en \LaTeX.
\end{document}
```

⁵Notemos que realizar un salto de línea entre las opciones no ocasiona ningún inconveniente y mejora la visualización del código.