

L^AT_EX

para tesis de ingeniería

```
\usepackage[spanish,mexico]{babel}  
  
\begin{portada}  
  
    \author{Carlos Ramos}  
  
    \title{\LaTeX{} para tesis de ingeniería}  
  
\end{portada}
```

Imágenes, ecuaciones, código, y referencias
Redacta tu tesis sin dolor de cabeza

L^AT_EX

para tesis de ingeniería

por Carlos Ramos



Leanpub

L^AT_EX para tesis de ingeniería

Carlos Alberto Ramos López

Este libro se encuentra a la venta en
<https://leanpub.com/tesis-en-latex/>

Esta versión se publicó el 5 de febrero de 2021.



Este es un libro publicado en la plataforma de **LeanPub**. Aunque este libro no fue generado mediante su sistema, incluyo esta información porque me parece una plataforma justa, que merece el crédito por apoyar a los autores con obras en progreso (o proclives a muchas actualizaciones, como lo son libros relacionados con tecnologías de la información).



Este trabajo está licenciado bajo **Creative Commons BY-NC-SA 4.0**, lo que implica que eres libre de compartir y redistribuir esta obra, siempre y cuando:

- Des el crédito de manera adecuada a la obra original con un enlace a la página <http://latex.ramoscarlos.com>.
- No hagas uso comercial de la obra (puedes compartirla gratis, pero no venderla).
- Si compartes la obra, lo hagas bajo esta misma licencia.

*Para Alberto, Lorenza, y Miriam,
cuyo fastidio continuo logró encaminarme
a esta adicción insaciable de leer y escribir.*

*Y a mi diosa maya, Ixchel,
porque aún no se da cuenta de las horas
que pasaré frente al ordenador escribiendo.*

Índice de figuras	XI
Índice de listados	XIII
Resumen	XV
1. Introducción	1
1.1. ¿Qué es?	2
Contenido sobre formato	2
Instrucciones	3
\TeX y \LaTeX	3
Infinidad de paquetes	4
1.2. Overleaf	5
1.3. Un primer documento	6
La instrucción <code>\maketitle</code>	8
La instrucción <code>\section</code>	8
Inicio y final de documento	9
Texto en español (el paquete <code>\inputenc</code>)	10
Clase del documento	11
1.4. ¿Compilar, recompilar?	11
Compila frecuentemente	12
1.5. Archivos temporales	12
2. Lo básico para redacción	15
2.1. El idioma español	15
2.2. Formato	16
Una nota sobre <code>\emph</code> y <code>\textit</code>	16

2.3.	Espacio en blanco en \LaTeX	16
2.4.	Texto entre comillas	19
	¿Y las comillas rectas?	19
2.5.	Escape de símbolos (el problema del guión bajo)	21
2.6.	Entornos	21
2.7.	Comentarios	22
2.8.	Listas	23
	Listas sin orden	23
	Listas ordenadas	23
	Listas anidadas	24
2.9.	Notas al pie	25
2.10.	Tamaños de fuente	25
2.11.	Secciones	26
	Párrafos y subpárrafos	27
	Secciones no numeradas	28
2.12.	Etiquetas y referencias	28
2.13.	Separación en archivos	29
2.14.	Estructura	30
3.	Figuras	33
3.1.	Evita referencias de lugar	33
3.2.	Figuras	34
	Indicador de ubicación	34
	Elementos flotantes	35
	Valores del indicador de ubicación	35
	Etiquetas en figuras	36
3.3.	El entorno <code>picture</code>	38
	Esquema de \LaTeX	40
	Figuras <code>.tex</code>	41
3.4.	Imágenes	42
	Ventajas de agregar por archivo	43
	Directorio de imágenes	44
3.5.	Opciones de la imagen	45
	<code>scale</code>	45
	<code>width</code>	47
	<code>height</code>	48
	<code>keepaspectratio</code>	48
	<code>angle</code>	49
	Entorno <code>sidewaysfigure</code>	49
3.6.	Advertencias de dimensiones	51

3.7. Subfiguras	51
4. Tablas	55
4.1. Generador de tablas	55
4.2. Entorno <code>table</code>	57
4.3. Entorno <code>tabular</code>	58
4.4. Generador: opciones adicionales	59
4.5. Bordes	61
4.6. Centrado de encabezados	63
4.7. Combinación de columnas	64
4.8. Combinación de filas	65
5. Ecuaciones	67
5.1. Ecuaciones en línea con el texto	67
5.2. Letras griegas y otros símbolos	68
5.3. Ecuaciones sin referencia	69
5.4. Llaves y paréntesis	69
5.5. Ecuaciones con referencia	70
5.6. Texto en entorno matemático	71
5.7. Casos	71
5.8. Ecuaciones multilínea	73
5.9. Entorno <code>split</code>	74
5.10. Entorno <code>align</code>	74
5.11. Evaluación de derivada	75
5.12. Ecuaciones lado a lado	76
5.13. Matrices	77
5.14. Más matemáticas	79
6. Más allá de lo básico	81
6.1. Citas	81
6.2. Columnas con <code>multicol</code>	82
6.3. Columnas con <code>minipage</code>	82
6.4. El preámbulo	84
6.5. Márgenes del documento	85
6.6. Colores	86
6.7. Enlaces	88
6.8. Texto resaltado	89
6.9. Definición de instrucciones	90
6.10. Redefinición de instrucciones	91
6.11. Separación de palabras	92

7. Código fuente	93
7.1. Paquete necesario	93
7.2. Traducción a español	93
7.3. Caracteres en español en el código	94
7.4. Opciones del paquete	95
7.5. Estilos	98
7.6. Código dentro de L ^A T _E X	100
7.7. Código de archivo externo	101
7.8. Resaltado de líneas	102
Un error en la Mac	104
7.9. Más parámetros sobre números de línea	105
8. Manejo de bibliografía	107
8.1. ¿Por qué las fuentes precisan otras instrucciones?	107
8.2. BIB _T E _X	109
8.3. Archivo de bibliografía	110
8.4. Tipos de fuentes bibliográficas	111
El formato general de una fuente	112
Atributos del tipo <code>article</code>	113
Atributos del tipo <code>inproceedings</code>	114
Atributos del tipo <code>book</code>	116
Atributos del tipo <code>masterthesis</code>	116
Atributos del tipo <code>misc</code>	117
8.5. ¿Por qué tanta llave?	118
8.6. ¿Nombre Apellido o Apellido, Nombre?	119
8.7. Múltiples autores y nombres con comas o “and”	119
8.8. Traducción	120
8.9. Otras opciones para compilar bibliografía	120
9. Secciones adicionales	123
9.1. Resultados y conclusiones	123
9.2. Tabla de contenido	124
9.3. Índices adicionales	125
9.4. Resumen (o <i>Abstract</i>)	125
9.5. Dedicatoria	126
9.6. Agradecimientos	127
10. Formato del documento	129
10.1. Espacio entre líneas	129
10.2. Encabezado y pie de página	130

10.3. Estilo de las leyendas	133
10.4. Estilo de páginas en blanco	134
10.5. Espacio de listas	134
10.6. La portada	135
10.7. Numeración de subsecciones	137
10.8. ¿Hasta qué nivel se muestra en el índice?	138
10.9. Estilo de los títulos de sección	138
La instrucción \titlespacing	138
La instrucción \titleformat	139
10.10. Estilo de capítulo	139
10.11. Fuentes	140
Comentarios finales	141
Agradecimientos	143
Bibliografía	145

Índice de figuras

1.1. Esquema de trabajo en L ^A T _E X.	3
1.2. Página principal de Overleaf.	5
1.3. Creando un nuevo proyecto en Overleaf.	6
1.4. Overleaf preguntando el nombre del nuevo proyecto.	6
1.5. Primer ejemplo en Overleaf.	7
1.6. Primer ejemplo en Overleaf, sin \maketitle.	8
1.7. Primer ejemplo en Overleaf, con una sección nueva.	9
1.8. Primer ejemplo en Overleaf, con error fuera del \begin{document}.	10
1.9. Botón de <i>Logs y archivos de salida</i> en Overleaf.	13
1.10. Botón de <i>Borrar archivos en la caché</i> en Overleaf.	13
2.1. Contenido del libro <i>Java All-in-One For Dummies, 5th Edition</i>	27
3.1. ¿Una figura vacía?	34
3.2. Una segunda figura vacía.	36
3.3. Tercera figura vacía.	37
3.4. Cuarta figura vacía.	38
3.5. Cuarta figura vacía, pero segunda leyenda.	38
3.6. Rectángulo vacío con picture.	38
3.7. Página principal de Overleaf, usando scale=1.	46
3.8. Página principal de Overleaf, al 50 % del ancho de línea.	47
3.9. Página principal de Overleaf, con width y height.	48
3.10. Página principal de Overleaf, con width, height, y keepaspectratio.	48
3.11. Página principal de Overleaf, rotada 90 grados.	50
3.12. Figura con dos subfiguras.	51
3.13. Tres gráficas de Octave.	52

4.1. Página principal del editor de tablas en línea.	55
4.2. Diálogo modal para nuevas tablas.	56
4.3. Contenido de nuestro primer ejemplo de tabla.	56
4.4. Código generado para nuestro primer ejemplo de tabla.	56
4.5. Opciones adicionales para la tabla.	59
4.6. Caja de texto para introducir leyenda y etiqueta.	60
4.7. Edición de bordes con editor en línea.	62
4.8. Editando código de tabla en ancho fijo, sin separar líneas.	64
8.1. Página de un artículo de la IEEE.	113
8.2. IEEE Xplore y su facilidad para exportar citas a BIB _T E _X	114
8.3. Artículo de conferencia disponible en Springer.	115
10.1. Título de capítulo con fncychap estilo Bjornstrup.	140

Índice de listados

1.1. Mi primer documento en \LaTeX	7
2.1. Fragmento de texto para demostración de espacio en blanco.	17
2.2. Espacio en blanco en \LaTeX	18
2.3. Uso de comillas en \LaTeX	20
2.4. Comentarios en \LaTeX	22
3.1. Código para una figura vacía.	34
3.2. Código para la segunda figura vacía.	37
3.3. Código para la tercera figura vacía.	37
3.4. Código para la cuarta figura vacía.	38
3.5. Código para crear un rectángulo con \LaTeX	39
3.6. Código de la figura 1.1.	41
3.7. Código del archivo esquema_latex.tex.	42
3.8. Código de la figura 1.2.	43
3.9. Uso de scale=1.	45
3.10. Uso de width.	47
3.11. Uso de width,height,y keepaspectratio.	48
3.12. Uso de angle=90	49
3.13. Figura en entorno sidewaysfigure.	49
3.14. Código para crear subfiguras.	52
3.15. Código para tres subfiguras.	53
4.1. Código de primera tabla generada.	57
4.2. Código para tabla con leyenda.	60
4.3. Código de tabla con bordes.	61
4.4. Código de tabla con bordes incompletos para filas.	61
4.5. Código de tabla con bordes incompletos para columnas.	62
4.6. Código de tabla de indicadores de ubicación.	63

4.7. Tabla con encabezado de cuatro columnas de ancho.	65
4.8. Tabla con columna de cuatro filas de alto.	66
5.1. Código para ecuación con referencia.	70
5.2. Código con texto usando <code>\mbox</code>	71
5.3. Uso del entorno <code>cases</code>	72
5.4. Sucesión con términos que desbordan una línea.	73
5.5. Sucesión en dos líneas.	73
5.6. Sucesión en dos líneas, centradas.	74
5.7. Entorno <code>align</code> para múltiples ecuaciones numeradas.	75
5.8. Uso de <code>minipage</code> para ecuaciones lado a lado.	77
5.9. Uso de <code>array</code> para matrices.	78
6.1. Estructura de archivos para incluir configuración en el preámbulo. . .	85
7.1. Instrucción en preámbulo para incluir caracteres especiales en código. .	94
7.2. Opciones globales para listados.	98
7.3. Estilo para lenguaje de programación \LaTeX	99
7.4. Estilo para mostrar errores de compilación de \LaTeX	99
7.5. Estilo para mostrar advertencias de compilación de \LaTeX	99
7.6. Código para el listado del estilo advertencias.	100
7.7. Uso de la instrucción <code>\lstinputlisting</code> para incluir archivos de código. .	102
7.8. Demostración de listado con línea resaltada.	102
7.9. Código para resaltar líneas.	103
7.10. Parámetro <code>linebackgroundcolor</code> con varias líneas resaltadas.	104
7.11. Listado con <code>firstnumber</code> y <code>stepnumber</code> definidos.	105
8.1. Entrada de un artículo de la IEEE.	114
8.2. Entrada de un artículo de Springer.	115
8.3. Entrada de \BibTeX de un libro.	116
8.4. Entrada de una tesis de maestría.	117
8.5. Entrada de dos recursos en línea.	117
9.1. Contenido completo de la dedicatoria.	127
10.1. Pie de página a la derecha de una página impar.	131
10.2. Pie de página a la izquierda de una página par.	132
10.3. Configuración de las leyendas con paquete <code>caption</code>	133
10.4. Lista enumerada con opción <code>nosep</code>	134
10.5. Código de la portada.	135

Este libro está diseñado con un público muy particular en mente, tal y como se establece en el título: estudiantes de ingeniería que deben escribir toda su tesis en este gran sistema de composición de textos que es \LaTeX (cuando lo leas en tu mente, que se escuche algo como *leitec*, no como material de preservativos).

Por supuesto, al empezar a trabajar con \LaTeX hace años esos no eran los adjetivos que utilizaba para describir este sistema, y más bien lo definía como “una herramienta tecnológica de tortura del siglo XX usada por la Inquisición Académica”.

Resulta que las imágenes no se insertaban al ser arrastradas al editor, y tampoco se quedaban donde yo les decía, ¿para qué servía el código entonces? Introducir caracteres en español era un horror, y los fragmentos de código que insertaba solo respetaban el alfabeto inglés, convirtiendo las letras acentuadas en símbolos horribles¹. Posiblemente, la mitad de mi tesis se fue en una pelea infructuosa en contra de \LaTeX .

Si tanta fue mi agonía, ¿por qué me empeñé tanto en usar \LaTeX como sistema de composición de un documento tan extenso como lo fue mi tesis? Por una razón: era requisito, al menos en mi carrera (un requisito no compartido uniformemente en mi institución, lo que solo sirvió para incrementar mi frustración en aquellos ayeres).

Quizá esta es tu primera exposición a \LaTeX , y has comenzado de una manera similar a la mía: sabrá Dios qué es, sabrá Dios cómo se usa², pero un director de tesis anticuado desea que redactes en esta cosa compleja, que ni interfaz tiene, que parece más código que otra cosa, y que no sabes por qué tienes que utilizar si existe Word.

Lo bueno es que estás de suerte porque este libro lo escribí como respuesta a “¿Qué hago si tengo que utilizar \LaTeX para escribir mi tesis y no sé dónde empezar?” Mi objetivo es evitarte muchas de las frustraciones que yo sufrí durante la redacción de mi tesis mientras aprendía a utilizar \LaTeX . Pero he de advertirte: esta obra contiene algunas de mis vivencias como tesista y algunas opiniones personales, incluidas con el objetivo de hacer amena la lectura. Toma lo que te sirva, desecha lo que no.

Definido el tono del libro, comencemos con la teoría.

¹Vale, no eran símbolos extraídos del *Necronomicón*, pero me causaron pesadillas por mi ignorancia.

²Como este es un documento laico, esta es la única referencia a una deidad que verás aquí.

1.1. ¿Qué es?

Si no te preguntas qué es \LaTeX , haces bien, ¿para qué saber? Ah, cierto, requisitos que cumplir³. Comencemos definiendo lo que *no es*. \LaTeX no es un sistema *WYSIWYG* (lo que ves es lo que obtienes⁴) como lo es Microsoft Word, a pesar de que ya hay opciones que permiten generar un documento de \LaTeX en una interfaz algo similar. No obstante, sigue siendo mejor escribir el “código” que será el documento final.

Pero, ¿por qué? Con la existencia de procesadores de texto tan poderosos, con una interfaz amigable, ¿para qué molestarse en aprender a utilizar \LaTeX , que aparenta ser tan arcaico? ¿Por qué nadie se ha preocupado en hacer una interfaz para que \LaTeX sea tan *WYSIWYG* como Word?

Esas respuestas solamente se puede obtener indagando en las características de \LaTeX , mismas que nos ayudarán a definir qué es.

Contenido sobre formato

\LaTeX se centra en el contenido. Si sabes un poco de páginas web, podríamos decir que \LaTeX es el equivalente al HTML de una página en el sentido de que define la estructura de cómo se debe desplegar el documento. Extendiendo la analogía, el código HTML no se puede visualizar sin un navegador que lo interprete. De la misma forma, \LaTeX necesita al compilador para pasar de código a documento presentable.

Aún más, podríamos decir que el CSS, la parte de una página que hace que se vea bonita, equivale a configuraciones de \LaTeX en el preámbulo, mismas que nos permiten modificar la presentación del contenido sin cambiar nada, o casi nada, del código utilizado durante la redacción.

En otras palabras, al redactar en \LaTeX no nos preocuparemos por el formato, al menos no en primera instancia, y posiblemente ni en segunda, ni en tercera. Es decir, no es necesario pensar en cómo se ven los títulos, la numeración de páginas, si el texto va justificado, o qué fuente vamos a utilizar, mientras estemos en la fase de redacción.

Tanto así que este libro empieza de lleno con el contenido, con enfoque en lo que vas a escribir, explicando cómo hacer listas numeradas o sin numerar, cómo meter imágenes, tablas o código, pero no brinda detalles de cómo cambiar cuestiones de estilo. Porque, de nuevo, importa más el contenido que el formato.

Además, si vas a redactar una tesis para tu institución educativa, lo más probable es que ya tengas una plantilla la cual tienes que seguir, a pesar de que la consideres un crimen en contra de todo sentido de la estética. Requisitos son requisitos.

No puedo terminar esta sección sin enfatizar nuevamente: el formato no es el protagonista en \LaTeX , ocúpate de escribir. Al último trabajaremos con la apariencia.

³Si estás aquí por gusto, eres de la inimaginable minoría. Igual será un placer ser tu guía.

⁴Del inglés “What You See Is What You Get”

Instrucciones

Aún no vemos cómo se ve el código de un documento realizado con \LaTeX , pero basta decir que se asemeja mucho a código de computadora, a cosas ininteligibles para seres mortales. Sí, suena a complicaciones innecesarias para realizar un documento que parece destinado a solo satisfacer el ego de unos cuantos.

No obstante, lo que distingue a \LaTeX es precisamente esta forma de trabajar, estas instrucciones que al final del día conforman el documento final, y que van mano a mano del punto anterior: el contenido va primero, antes que todo formato.

¿Y qué es una instrucción? Es una orden o comando que la computadora tomará para traducirla desde tu “yo escribo contenido” al producto final en PDF⁵ de “computadora genera formato”. Podemos verlo en términos de la figura 1.1:

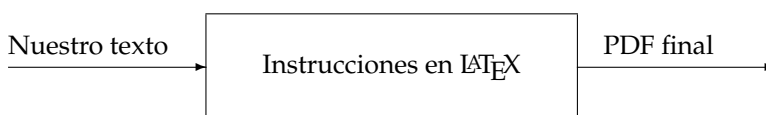


Figura 1.1: Esquema de trabajo en \LaTeX .

Como ejemplo, para generar este libro solo pensé en este capítulo como “Introducción”. Al momento de escribir, no sabía si sería “Capítulo 1”, “Capítulo I”, o solo diría “1”, o si estaría numerado, siquiera.

No me preocupé de si estaría alineado a la derecha o a la izquierda, o el tamaño de la letra. Simplemente introduje el siguiente código (técnicamente, es más correcto decir “la siguiente instrucción” pero suena mejor de esta manera):

```
\chapter{Introducción}
```

Y con eso me liberé de preocuparme por el formato. Sí, en algún momento el cómo regresó a perseguirme, pero no mientras estaba escribiendo e investigando.

De la misma forma, no deberás ocuparte del formato mientras estés en la fase de ~~generar resultados falsos y rellenar con mentiras para llegar a un mínimo de hojas~~ investigación, desarrollo, y documentación, sino hasta que tu asesor te diga que el contenido está bien aunque el formato no tanto.

\TeX y \LaTeX

En el punto anterior vimos que \LaTeX utiliza instrucciones para generar documentos. Pero, en realidad, \LaTeX no es otra cosa que \TeX con esteroides. ¿Qué significa ésto? ¿Acaso existen otras versiones? ¿Qué importa si es \LaTeX , \TeX , \LuaTeX , o sabrá que otra abominación con \TeX al final?

⁵Se puede realizar una conversión a otros formatos, pero eso escapa el alcance de este libro.

No es que importe mucho en sí la historia, pero un tío llamado Donald Knuth creó \TeX allá por 1978, con una licencia de código abierto. De ahí, otras personas empezaron a empalmar sus modificaciones y atajos, como más formas de estilizar texto, o para sus aplicaciones particulares de fórmulas, código, alfabetos, y más.

Entonces, \LaTeX viene a ser la forma más popular de \TeX que existe, con macros o instrucciones agregadas por muchísima gente a lo largo de ya más de cuarenta años.

Por tomar otra analogía de los chicos de sistemas, \LaTeX vendría a ser como Ubuntu, la versión de Linux más conocida, y \TeX algo así como Debian, el papá de los pollitos. Y todas las demás variantes, que encontrarás muchas, son distintos sabores de \TeX para aplicaciones particulares.

¿Y eso qué tiene que ver con nosotros? Bueno, es hora de colocar el último clavo sobre el ataúd de la definición de \LaTeX .

Infinidad de paquetes

Ahora que hemos hablado de las instrucciones, como `\chapter`, y sabemos que \LaTeX es una versión aumentada de \TeX gracias a contribuciones de muchas personas, podemos entender la definición de \LaTeX (según Wikipedia [1]):

\LaTeX es un sistema de composición de textos que está formado mayoritariamente por órdenes construidas a partir de comandos de \TeX .

Con lo que hemos visto podemos construirnos una definición de mayor utilidad:

\LaTeX es una herramienta quimera, nacida de \TeX , que ha ido creciendo gracias a contribuciones de personas a lo largo de cuarenta años, cuyo objetivo es que podamos crear un documento que luce espectacular, enfocándonos mayormente en el contenido.

Por supuesto, ese objetivo muchas veces se pierde entre tantas instrucciones y sabores. ¡Más de cuarenta años de personas agregando paquetes sin orden ni concierto! Sin una guía, es difícil comprender cómo podemos lograr nuestro objetivo de la manera más sencilla, sin que una instrucción afecte a otra.

Además, esta misma infinidad de paquetes hace que un editor *WYSIWYG* sea poco factible, ¿cómo hacerlo si hay miles de paquetes que pueden modificar el comportamiento de las instrucciones? Sí, claro, el entorno *itemize* es una lista, por ejemplo, ¿pero cómo debe verse? Volvemos al mismo punto: importa más el hecho de que sea una lista, independientemente del cómo se ve. Por lo tanto, no tiene mucho sentido ver el resultado aproximado en tiempo real (como en Word) sino que podemos esperar para ver el resultado visual tras la fase de compilación.

¿Compila-qué? No te preocupes, ya llegaremos a ello en este mismo capítulo. Por lo pronto, ya tenemos una definición de \LaTeX y sabemos más o menos para qué nos sirve. Es tiempo de realizar un primer ejemplo en una plataforma en línea.

1.2. Overleaf

En mis tiempos (**tose, tose**) no existía una plataforma en línea en la cual realizar un documento en \LaTeX . Ni siquiera era universal el Javascript, mucho menos se podía concebir un equipo que compilara en tiempo real y permitiera generar un PDF al vuelo de varios archivos de código.

Dejando de lado la nostalgia de un viejo, el punto es que existe una plataforma gratuita en línea que permite escribir documentos complejos de \LaTeX , sin problemas ni peros (bueno, si quieres colaborar con alguien o compilar desde Dropbox o GitHub sí hay costo, pero estamos empezando y no haremos nada así a lo largo de este libro). Dicha plataforma es **Overleaf**⁶, cuya página principal luce similar a la figura 1.2.



Figura 1.2: Página principal de Overleaf.

En la figura 1.2 se muestra el formulario de registro, el cual solo pide correo electrónico y contraseña, aunque también te puedes registrar mediante tu cuenta de Google. Una vez estés registrado, podemos empezar a crear el documento de prueba.

⁶Por si ves esto impreso, el sitio es <https://www.overleaf.com/>

1.3. Un primer documento

Dado que un documento de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ puede estar compuesto por muchos archivos, como después veremos, lo primero que hay que hacer es crear un proyecto (alias sofisticado de “directorio”) en Overleaf, con *Nuevo proyecto* \rightarrow *Proyecto vacío*, como muestra la figura 1.3. Aunque podemos crear proyectos a partir de ejemplos o de plantillas, empezaremos desde cero.

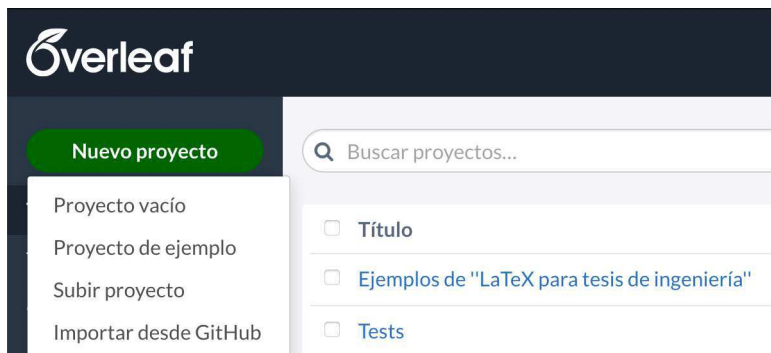


Figura 1.3: Creando un nuevo proyecto en Overleaf.

Al dar clic en la opción *Proyecto vacío* surgirá una pantalla modal, como la figura 1.4, que preguntará por el nombre del proyecto. En mi caso, colocaré el nombre de *Ejemplos de "LaTeX para tesis de ingeniería"*.

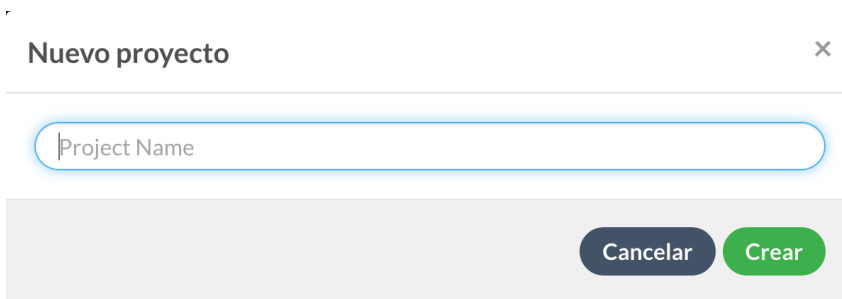


Figura 1.4: Overleaf preguntando el nombre del nuevo proyecto.

Si todo sale bien, se desplegará una pantalla con información similar a lo que se muestra en la figura 1.5, solo con un tamaño de fuente más pequeño. Obviamente el nombre del proyecto para ti será diferente (espero), así como el autor asignado de manera predeterminada. No obstante, en la columna izquierda debes tener un archivo llamado `main.tex`, en la columna central el contenido de dicho archivo, mientras que el lado derecho muestra el resultado en PDF con el nombre de tu proyecto, tu nombre como autor, y la fecha.

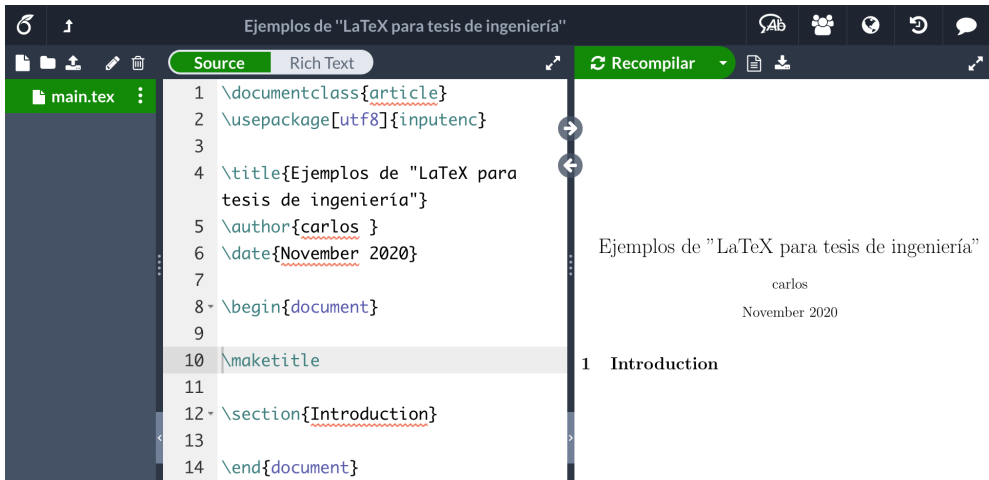


Figura 1.5: Primer ejemplo en Overleaf.

Para empezar a comprender cómo funciona \LaTeX , revisemos el código del listado 1.1. En sí, lo que se puede deducir del PDF generado (lado derecho de la figura 1.5) es que las líneas 4, 5, y 6 tienen un reflejo directo en el resultado final: se muestra el título del documento, el autor, y la fecha. También la línea 12 aparece en el documento, aunque el texto “Introduction” está precedido por un número.

¿Cómo es que esas cuatro líneas aparecen, una tras otra, pero en el código son separadas por la línea 8 y 10? ¿Y de dónde sale ese número asignado al texto en negritas de “Introduction”? Y, claro, ¿por qué aparecen las comillas de cierre tanto al inicio como al final en nuestro título?

```

1 \documentclass{article}
2 \usepackage[utf8]{inputenc}
3
4 \title{Ejemplos de "LaTeX para tesis de ingeniería"}
5 \author{carlos }
6 \date{November 2020}
7
8 \begin{document}
9
10 \maketitle
11
12 \section{Introduction}
13
14 \end{document}

```

Listado 1.1: Mi primer documento en \LaTeX .

La instrucción `\maketitle`

Empezaremos a responder las preguntas de manera empírica, eliminando la línea 10 del listado 1.1, que corresponde a la instrucción `\maketitle`. Se presiona el botón de *Recompilar* y se observa el nuevo resultado, tal como muestra la figura 1.6.

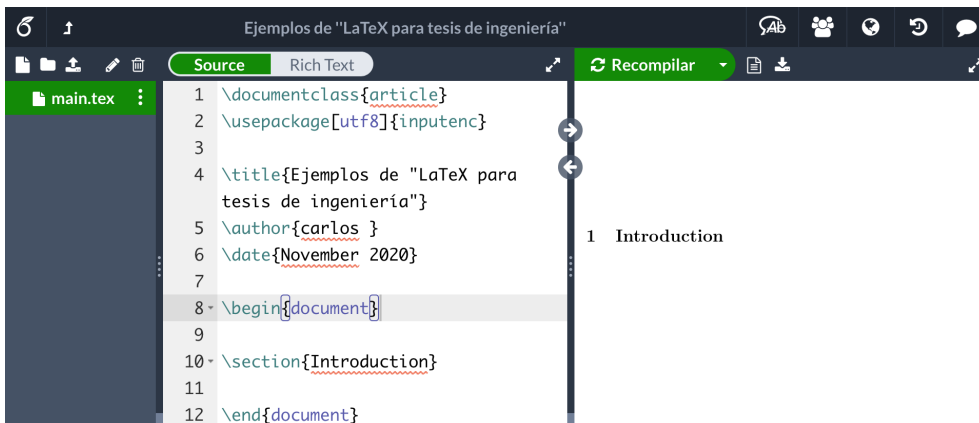


Figura 1.6: Primer ejemplo en Overleaf, sin `\maketitle`.

Dado que al eliminar la instrucción `\maketitle` ya no podemos ver en el PDF ni el título, ni el autor, ni la fecha, podemos asumir que esa instrucción se encargaba de mostrar esos tres datos centrados en el documento.

Y, basados en la misma información, podemos deducir que las instrucciones `\title`, `\author`, y `\date` son instrucciones para almacenar valores (ya que no imprimen en el documento), algo así como variables de \LaTeX , que la instrucción `\maketitle` puede leer e imprimir.

¿Importa cómo? No, solamente que lo hace. Debemos recordar que no tenemos motivos para preocuparnos por el formato o funcionalidad mientras se presente la información correcta en el documento.

Ya que sabemos los efectos de eliminar `\maketitle`, regresamos la línea a donde pertenece antes de continuar.

La instrucción `\section`

La línea 12, `\section{Introduction}`, imprime “Introduction” precedido por un uno. Por ello, podemos asumir que \LaTeX , de alguna manera, enumerará consecutivamente las secciones. Haremos dos cosas para probar esta teoría:

1. Cambiar el texto de “Introduction” a “Introducción”.
2. Crear una nueva sección llamada “Esta sección debería llevar el número dos...”

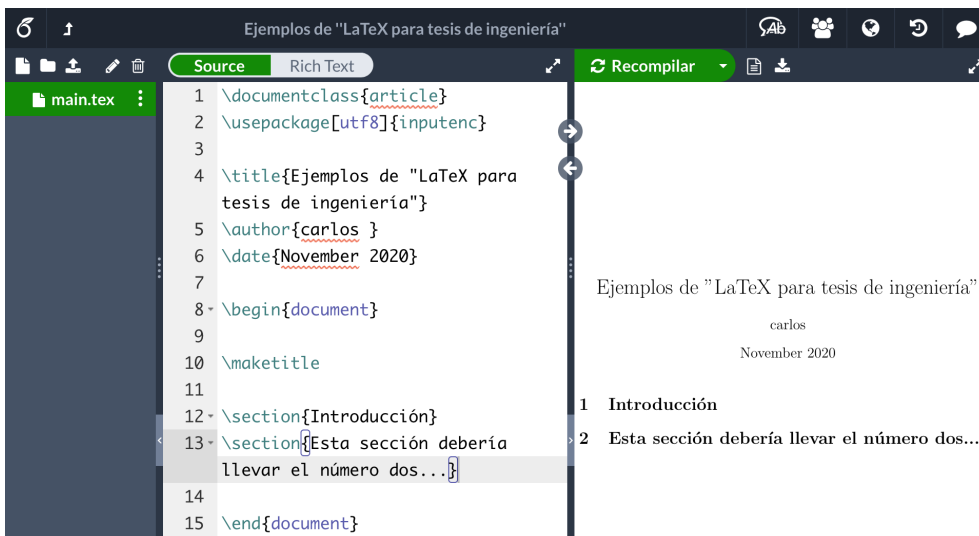


Figura 1.7: Primer ejemplo en Overleaf, con una sección nueva.

La figura 1.7 refleja los cambios (tras volver a presionar el botón *Recompilar*), y confirma las sospechas: \LaTeX agregará un consecutivo a cada sección, de manera automática, sin nuestra intervención (de nuevo, así nos preocupamos por la estructura del documento, no del cómo se ve). La líneas de código editadas fueron:

```
\section{Introducción}
\section{Esta sección debería llevar el número dos...}
```

Inicio y final de documento

Siguiendo con el listado 1.1, las líneas 8 y 14 marcan el inicio y final del documento. ¿Esto qué quiere decir? Que todo lo que *se ve* en el documento va entre estas dos instrucciones, por lo que si intentamos ejecutar el `\maketitle` antes de estas instrucciones, posiblemente se genere un error (y sí, la figura 1.8 muestra el error que ocurre tras *Recompilar*).

De nuevo, no es nuestra intención conocer el cómo \LaTeX hace lo que hace, pero si algo va destinado a imprimirse o verse en pantalla, va entre `\begin{document}` y `\end{document}`. ¿Qué es lo que va fuera? Como en nuestro ejemplo, la declaración de “variables” como el autor, el título, y la fecha. También, la inclusión de paquetes que extienden las capacidades de \LaTeX .

En este primer documento ya se incluye el paquete `inputenc`.

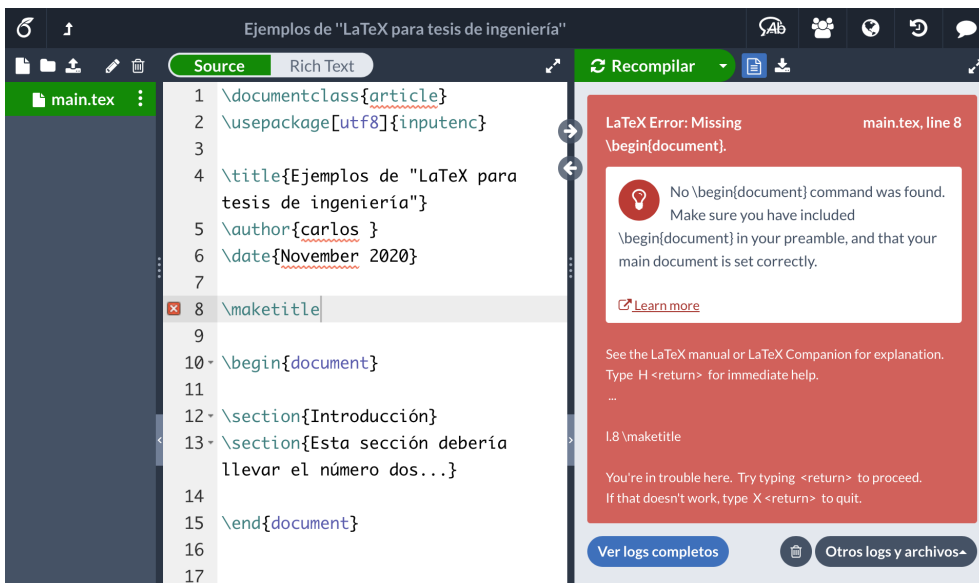


Figura 1.8: Primer ejemplo en Overleaf, con error fuera del `\begin{document}`.

Texto en español (el paquete `\inputenc`)

La instrucción `\usepackage` permite usar los paquetes que otra persona creó. Es decir, permite usar las instrucciones que alguien más ya hizo y decidió hacer públicas. Pero, en un ejemplo tan sencillo, ¿para qué necesitamos cargar un paquete?

Cuando Donald Knuth creó \TeX , lo más probable es que pensara en sí mismo, sus fines, y su idioma (el inglés). Naturalmente, los acentos y las “ñ” no le pasaban por la mente, por lo que no son soportados de manera nativa. Para poder insertar los caracteres propios del idioma español, sin que estos sean convertidos a símbolos raros, se agrega el paquete `inputenc`, que permite definir en qué idioma estamos escribiendo (lo formal es decir algo como “define la codificación de los caracteres de entrada”, pero se entiende menos).

Y una codificación⁷ muy utilizada, y que contiene los caracteres del español, es la UTF-8. No entraremos en detalles técnicos de representación, solo diremos que sin el paquete `inputenc` con la codificación `utf8` no se ven los caracteres en español de manera adecuada.

Como dato meramente informativo: este paquete no se incluye solamente porque tu Overleaf se haya configurado en español. El código ASCII quedó atrás hace mucho tiempo, siendo el UTF-8 el nuevo estándar de codificación. Así que, aunque tu documento solo utilice caracteres del idioma inglés, se recomienda utilizar el paquete UTF-8 (razón por la que Overleaf lo carga de manera predeterminada).

⁷En inglés, *encoding*.

Clase del documento

Dejamos la primera línea para el final, donde se define la clase del documento. Hay varias clases de documento, y muchas más si contamos las plantillas como el libro Tufte, pero a lo largo de este libro utilizaremos artículos (clase `article`) y libros (clase `book`).

Los artículos serán utilizados para hacer las demostraciones cortas, como el listado 1.1, y el libro es la base de este documento, del cual tienes todo el código fuente a tu disposición.

Aunque existen muchas diferencias entre ambas clases, una de las principales es la estructura, o la forma en la que se subdivide, como se muestra en las siguientes listas:

Libro

- Capítulo
 - Sección
 - Subsección

Artículo

- Sección
 - Subsección

Dicho de otra forma, un libro tiene más subdivisiones porque cuenta con separación por capítulos mientras que un artículo define su nivel principal a través de secciones.

La otra diferencia es que un artículo coloca el título en la primera hoja, seguido de todo el contenido (lo cual vimos en las figuras 1.6 y 1.7), mientras que un libro crea una portada y comienza el contenido principal a partir de la segunda hoja (como en este mismo documento).

Al experimentar con la clase `book` solo ten en cuenta que si no tienes un capítulo (`\chapter`), las secciones estarán numeradas como “0.1” y “0.2”, por la falta del nivel superior.

1.4. ¿Compilar, recompilar?

Al momento de regenerar el documento dimos clic al botón *Recompilar*, el cual hace la magia que actualiza el documento PDF. Pero ¿qué es? ¿qué hace? ¿por qué tenemos que presionarlo para ver cada cambio?

Dentro de las características de \LaTeX se mencionó que no era un editor WYSIWYG, lo que implica que al escribir no veremos lo recién escrito en la forma que aparecerá en el documento PDF. Y, si recuerdas la figura 1.1, hablamos de intrucciones que nos ayudaban a enfocarnos en el contenido, y que esas instrucciones luego se convertían en el documento con formato.

Pero, ¿cómo es que esas instrucciones pasan de lo que escribimos al documento final? El verbo que buscamos es *compilar*. Aquí lo definiremos como:

La traducción de las instrucciones **válidas** de \LaTeX a un documento hermoso, digno de ser mostrado como tesis.

Es importante el énfasis en la palabra “válidas”. En Word, si haces algo que no le gusta, desacomoda todo el documento, cosa que arreglas con un *Ctrl + Z*. En \LaTeX , si haces algo que no le gusta, no tendrás documento de salida. Nada, cero, inexistente... como tu calificación (tú no te preocupes, para eso tienes este libro).

Siguiendo con el vocabulario: si ya compilaste una vez, la siguiente vez será una *recompilación* (por eso Overleaf utiliza *recompilar*, porque automáticamente genera el PDF la primera vez). A lo largo de este documento no se hace distinción entre ambas acciones: *compilar* y *recompilar* se usan de manera intercambiable.

Compila frecuentemente

Antes de continuar, aventurero, escucha mi advertencia: recuerda compilar frecuentemente. ¡Compila, compila! ¿Escribes una línea? Compila. ¿Escribes un párrafo? Compila. ¿Agregas una nueva imagen? Compila. ¿Cambiate un acentito? Compila.

Créeme, cualquier cosa insospechada puede causar error. Luego, cuando tengas todo un capítulo escrito pero no puedas ver nada mas que errores sobre errores, odiarás \LaTeX . Solo confía en mí: compila con la misma paranoia con la que guardas el documento.

1.5. Archivos temporales

En esta obra no veremos cómo es que \LaTeX hace su magia (bueno, un poco, y hasta el final), pero hay que responder a la pregunta: ¿dónde pone \LaTeX todo lo que necesita para pasar de nuestro texto plano al PDF final?

Después de todo, requiere mantener un contador de en qué capítulo y en qué sección va, traducir comandos a texto dispuesto en el PDF, obtener los valores de variables como el autor o la fecha, entre otras fechorías más.

Debido a todo lo que tiene que hacer, el proceso de compilación genera archivos temporales para ayudarlo a llegar del texto plano al documento final. Algunas de las extensiones de los archivos generados en el proceso son: *.toc, *.lof, *.bbl, *.aux, entre otras.

No ocupamos saber para qué requiere tanto archivo, mientras haga bien su trabajo... y es ahí donde viene un problema. Cuando nosotros cometemos un error, e indefectiblemente lo haremos, el compilador no puede acabar su tarea. ¿Y eso qué? Volvemos a iniciar el proceso de compilación y ya, ¿no?

Sí... y no. Como al compilador le gusta ser eficiente, si unos archivos no han presentado cambios entonces no los reescribe con el fin de ahorrar tiempo, lo que puede terminar con un error que se ha quedado guardado en un archivo temporal.

Dicho de otro modo, podemos tener errores fantasma que ya fueron solucionados en el código fuente pero que se han quedado atorados en los archivos temporales. Y es por esta razón que es necesario hablar de ellos: debes saber cómo eliminarlos para cuando tengas un error persistente, para descartar la posibilidad de que el error solo exista en un archivo temporal.

Overleaf nos da la opción de eliminar los archivos temporales, aunque no hay un botón de tan fácil acceso como *Recompilar*. Para eliminar los archivos primero tenemos que ubicar el botón de *Logs y archivos de salida*, que se encuentra justo al lado derecho del botón *Recompilar*, como que se muestra en la figura 1.9.



Figura 1.9: Botón de *Logs y archivos de salida* en Overleaf.

Al dar clic sobre dicho botón se abrirá un pequeño menú, como se muestra en la figura 1.10, que nos da acceso al botón *Borrar archivos en la caché*, con el ícono de un bote de basura, que nos permite deshacernos de los archivos temporales asociados al proceso de compilación de nuestro proyecto o archivo.



Figura 1.10: Botón de *Borrar archivos en la caché* en Overleaf.

Resumen

En este capítulo definimos qué es \LaTeX de una manera que facilita su interacción con el sistema e indagamos empíricamente qué hace cada línea del primer documento generado en la plataforma en línea Overleaf.

Además, establecimos dos consejos que te ayudarán mucho en tu aventura de terminar tu tesis con \LaTeX : compila frecuentemente y elimina archivos temporales después de una compilación fallida.

Armados con este conocimiento, estamos listos para empezar a ver cómo usar negritas, cursivas, y otras cosas de formato en \LaTeX .