

Introducción breve a L^AT_EX

José Cuevas Barrientos

<https://github.com/JoseCuevasBtos/apuntes-tex>

4 de agosto de 2020

Índice

1. Haciendo un documento básico desde cero	1
1.1. “Hola mundo”	1
1.2. Formato de texto	4
1.3. Notas, listas, figuras y tablas	6
1.4. Bibliografías	11
1.5. Configurar mi documento	12
2. Escribir matemáticas	14
2.1. Básico	14
2.2. Ecuaciones numeradas, alineadas, matrices y más	20

1. Haciendo un documento básico desde cero

En primera vamos a ver la estructura y características generales de L^AT_EX. A diferencia de MS Word, L^AT_EX no es un generador WYSIWYG¹, sino que se escribe mediante texto en un archivo de texto y un lenguaje de marcado. No os confundas, L^AT_EX **no** es un lenguaje de programación, y no se parece a uno, sino que es como escribir texto común con comandos entre medio.

1.1. “Hola mundo”

Los comandos suelen tener esta estructura `\comando{opciones}` donde la información obligatoria para el comando suele ir entre llaves `{}` a los que

¹en. *what you see is what you get*; lo que ves es lo que obtienes.

llamaremos *campos obligatorios*, mientras que la información opcional suele ir entre corchetes []. Todo documento comienza así:

```
\documentclass[opciones]{tipo de documento}
```

Por el momento en las opciones incluiremos el tamaño normal de letra que es 10, 11 o 12pt y el tipo de documento será `article` o `book` según lo que se desee escribir.

Luego viene un comando de la forma:

```
\begin{document}  
Hola mundo.  
\end{document}
```

Varios comandos tendran la misma estructura, a estos les llamaremos *entornos* y a la parte de `document` su “tag”. El entorno `document` señala todo lo perteneciente al contenido del archivo. Todo lo que esté fuera de este entorno (en particular antes de él) le diremos la *cabecera* del documento. Debería compilar este archivo básico para corroborar que su instalación de L^AT_EX ha sido exitosa. Varios programas (como TeXMaker) tienen compiladores incluidos, pero si se interesa en hacerlo por la terminal puede ver el apéndice.

Paquetes. Igual que los lenguajes de programación permiten añadir módulos para añadir funciones (como `<stdio.h>` en C, o `numpy` en Python), L^AT_EX ocupa paquetes que se incluyen en la cabecera, de momento añadiremos dos:

```
\usepackage[spanish]{babel}  
\usepackage[utf8]{inputenc}
```

El primero le indica a L^AT_EX que estamos escribiendo en español, para traducir texto del documento como escribir “Capítulo” en lugar de “Chapter”. El segundo le dice a L^AT_EX que lea las tildes y otros caracteres especiales (e.g. á, ñ, ç, etc.) sin problema.

Título. (Casi) todo documento tiene cosas como título, autor y fecha, para incluirlos en L^AT_EX debe ocupar correspondientemente los siguientes comandos en la cabecera:

```
\title{Mí título interesante}  
\author{Yo}  
\date{\today}
```

Nótese que en la fecha utilice el comando `\today`, este lee la fecha del día en el computador y el paquete de `babel` lo traduce al español, pero puede ocupar cualquier otra que se le antoje. Dentro del documento y antes del resto del texto escriba este comando para generar el título estándar de L^AT_EX:

```
\maketitle
```

Secciones. Todo texto suele dividirse en secciones. \LaTeX también, y además las enumera automáticamente. En un artículo la mayor división es un sección, luego una subsección y luego una sub-subsección:

```
\section{Esta es mi sección}
\subsection{Esta mi subsección}
\subsubsection{Esta mi sub-subsección}
```

Un libro también admite todos los comandos anteriores, pero la mayor distinción es un capítulo `\chapter{...}` y se pueden agrupar los capítulos en partes `\part{...}`, donde las últimas se enumeran con números romanos; pero a diferencia de los capítulos no son obligatorios.

Cortes de línea. Cuando hay un salto entre punto aparte y el otro párrafo eso se dice un *corte de línea*, lo que uno está acostumbrado a hacer con la tecla **Enter** de vuestros teclados. No obstante en el código de \LaTeX puede notar que no genera efecto alguno, esto se debe a que en los lenguajes de marcado se da esta precaución para evitar líneas excesivamente largas en el código. Para hacer este corte hay varias formas: `\` suele ser el más común, y no genera sangría en la línea contigua. Para hacer un corte con sangría puede usar dos veces **Enter** en el código. Un efecto similar es el de usar `\par`. La sangría se puede añadir con `\indent` y quitar con `\noindent` de ser necesario. Por último, si necesita forzar un corte de línea, para evitar la sobrecarga de caracteres por ejemplo, puede usar `\break`.

```
Texto de ejemplo.\break
Texto de ejemplo.\
Texto de ejemplo.\par
Texto de ejemplo.
```

Texto de ejemplo.
Texto de ejemplo.
Texto de ejemplo.
Texto de ejemplo.

Caracteres especiales. Sabemos que los comandos se inician con `\`, por ende, ¿cómo se escribe dicho caracter de ser necesario en \LaTeX ? Esta clase de caracteres se dicen *especiales* pues cumplen una función por si solos y se pueden escribir así:

\backslash	\backslash	$_$	$-$
$\%$	$\%$	$\#$	$\#$
$\$$	$\$$	$\&$	$\&$
$\{\}$	$\{\}$		

Además debe tener en consideración que las comillas en L^AT_EX son también distintas para escribir algo “así” se requiere ‘‘así’’. Donde las dos primeras son tildes graves y las últimas son apostrofes.

1.2. Formato de texto

Estilos. Uno suele cambiar el formato, e.g. a **negritas**, o *cursivas*, para ello L^AT_EX ocupa:

<code>\textup{Derecho}</code>	<code>\upshape</code>	Derecho
<code>\textit{Cursivas}</code>	<code>\itshape</code>	<i>Cursivas</i>
<code>\textsl{Inclinado}</code>	<code>\slshape</code>	<i>Inclinado</i>
<code>\textsc{Versallitas}</code>	<code>\scshape</code>	VERSALLITAS
<code>\textmd{Mediano}</code>	<code>\mdseries</code>	Mediano
<code>\textbf{Negritas}</code>	<code>\bfseries</code>	Negritas
<code>\textrm{Romano}</code>	<code>\rmfamily</code>	Romano
<code>\textsf{Sans-serif}</code>	<code>\sffamily</code>	Sans-serif
<code>\texttt{Máquina}</code>	<code>\ttfamily</code>	Máquina

El segundo tipo de comando es lo que se dice un *modificador*, al usarlo modifica todo el entorno², e.g., `{\scshape Texto de Ejemplo}` TEXTO DE EJEMPLO.

Tamaños. Para los tamaños de letra puedes usar un modificador o un entorno, ambos ocupan el mismo tag:

`tiny` `scriptsize` `footnotesize` `small` `normalsize`
`large` `Large` `LARGE` `huge` `Huge`

Cambiar el tamaño de letra por defecto (de 10 a 12pt por ejemplo) afecta también los otros tamaños de letra.

Alineación. L^AT_EX permite de dos formas el cambio de alineación de texto:

Alineación	Entorno	Modificador
Izquierda	<code>flushleft</code>	<code>\raggedright</code>
Derecha	<code>flushright</code>	<code>\raggedleft</code>
Centro	<code>center</code>	<code>\centering</code>

²En realidad modifica el entorno restante, ya que el contenido dentro del mismo entorno que viene ántes del modificador no se ve afectado.

Colores. Para admitir colores extra se debe importar el siguiente paquete en la cabecera:

```
\usepackage{xcolor}
```

Para usarlo puedes escribir: `{\color{red} texto en rojo}` **texto en rojo** o `\textcolor{blue}{texto en azul}` **texto en azul**. Los nombres de colores por defecto son:

black	darkgray	lime	pink	teal
blue	gray	magenta	purple	violet
brown	green	olive	red	white
cyan	lightgray	orange		yellow

Para poner colores en el fondo se utiliza `\colorbox{cyan}{Así}` **Así**. Además puedes utilizar un porcentaje de un color, por ejemplo, si sólo queremos usar el 50 % de negro podemos escribir `\colorbox{black!50}{esto}` **esto**, y si queremos usar 50 % azul y el resto rojo `\colorbox{blue!50!red}{haga esto}` **haga esto**.

Por sobre eso, usted puede definir nuevos colores en la cabecera:

```
\definecolor{ejemplo}{HTML}{00c89c}
```

Donde lo de HTML indica que la forma de definir colores es mediante el llamado código hexadecimal o **hex** para acortar. Este suele ser el método más común de definir colores, en línea lo reconocerá pues ocupan el prefijo '#', e.g., **#00c89c**.

Varias columnas. Si usted quisiera, de forma global, tener un documento escrito en dos columnas, sólo basta agregar la opción `twocolumn` en `\documentclass`. No obstante, si desea hacerlo localmente, esto es, en una sola parte del documento, debe importar el paquete:

```
\usepackage{multicol}
```

y luego usar el entorno `multicols` seguido del número de columnas:

```
\begin{multicols}{2}
    Muchos años después, frente al pelotón de
    fusilamiento, el coronel Aureliano Buendía...
\end{multicols}
```

Muchos años después, frente al pelotón de fusilamiento, el coronel Aureliano Buendía había de recordar aquella tarde remota en que su padre lo llevó a conocer el hielo. Macondo era entonces una aldea de veinte ca-

sas de barro y cañabrava construidas a la orilla de un río de aguas diáfanas que se precipitaban por un lecho de piedras pulidas, blancas y enormes como huevos prehistóricos.

1.3. Notas, listas, figuras y tablas

Notas. L^AT_EX admite dos tipos comunes de notas: al pie de la página y al margen de ella. La primera se especifica con el comando `\footnote{Esta es una nota al pie de página.}`³ mientras que las notas de margen se hacen con `\marginpar{Y ésta, una nota al margen.}`. Para revertir el lugar se puede usar `\reversemarginpar{...}`.

Para más información y mejor manejo de las notas al margen se recomienda revisar el paquete `marginnote`.

Listas. Se dividen en tres: numeradas, no numeradas y descripciones (similar a un glosario o diccionario). Todas se definen por un entorno y sus elementos se diferencian por el comando `\item`. Para una lista enumerada el entorno utiliza el tag `enumerate`, las no numeradas el tag `itemize` y las descripciones `description`. En las descripciones, la palabra (o frase) de título se denota en el campo opcional de `\item[Aquí...]`. Por ejemplo:

Y ésta, una nota al margen.

```
\begin{enumerate}
  \item Naranjas.
  \item Manzanas:
    \begin{itemize}
      \item Rojas.
      \item Verdes.
      \item Amarillas.
    \end{itemize}
  \item Bananas.
\end{enumerate}
\begin{description}
  \item[Fruta] Fruto comestible de ciertas plantas
    cultivadas; e.g., la pera, la guinda, la fresa, etc
  .
  \item[Verdura] Hortaliza, especialmente la de hojas
    verdes.
\end{description}
```

1. Naranjas.

³Esta es una nota al pie de página.

2. Manzanas:

- Rojas.
- Verdes.
- Amarillas.

3. Bananas.

Fruta Fruto comestible de ciertas plantas cultivadas; e.g., la pera, la guinda, la fresa, etc.

Verdura Hortaliza, especialmente la de hojas verdes.

Figuras y tablas. Por defecto, L^AT_EX no permite importar imágenes al documento por lo que se requiere importar el paquete:

```
\usepackage{graphicx}
```

Luego para importar la figura (asumiendo que la imagen está en la misma carpeta que el archivo) se utiliza el siguiente comando:

```
\includegraphics[opciones]{imagen.jpg}
```

En las opciones usualmente van especificaciones sobre el tamaño, verá que sin ella la figura se importa al máximo tamaño para el cual no pierde detalle, para importarlo digamos a la mitad de su tamaño real puede usar `scale=.5`, pero usualmente es mejor especificar directamente ya sea el largo (`width`) o el alto (`height`) del archivo que debe hacerse en `cm`, `mm` o `in` (pulgadas). También podemos decirle que sea tan larga como el tamaño del texto usando `width=\textwidth`.

Para que la imagen este apartada del texto se incluye todo dentro de un entorno `figure` cuyo único propósito es ese de decirle a L^AT_EX que trate la imagen individualmente. Dentro de él se recomienda centrar el texto (o en este caso el contenido) y al final del entorno se recomienda añadir alguna forma de corte de línea para mejorar el formato.

Posicionamiento. Para saber donde ubicar la imagen, L^AT_EX utiliza los siguientes símbolos para abreviar:

Descripciones. Además las imágenes poseen descripciones cortas, estas pueden agregarse en L^AT_EX dentro de un entorno `figure` con el comando `\caption{Mi descripción.}`, veamos un ejemplo donde se utiliza todo lo anterior:

-
-
- h Aproximadamente en el lugar donde se ubica en el código.
 - t En el tope de la página siguiente.
 - b En el fondo de la página actual.
 - p En la siguiente página, destinada exclusivamente para figuras.
 - ! Fuerza la posición indicada.
-
-

```
\begin{figure}[%h]
  \centering
  \includegraphics[width=10cm]{fractal.png}
  \caption{Un fractal}
\end{figure}
```

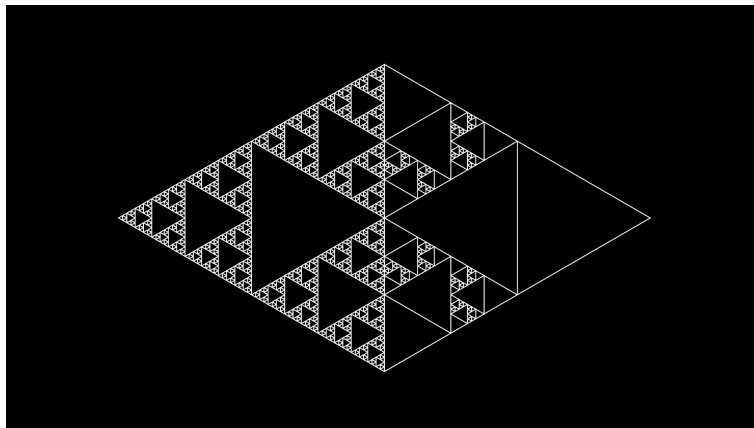


Figura 1: Un fractal

Tablas. Las tablas se definen así:

```
\begin{tabular}{formato}
  ...
\end{tabular}
```

Donde el *formato* se define por lo siguiente:

Los marcados por † requieren del paquete `array`. Las celdas en una columna se separan por el símbolo `&` y las columnas se separan por un corte de línea. Además `\hline` genera una línea horizontal lo que puede ser útil. Tal como uno pone las figuras en un entorno `figure`, existe el entorno

l	Celdas hacia la izquierda.
c	Celdas centradas.
r	Celdas hacia la derecha.
p{longitud}	Párrafo de cierta longitud.
m{longitud}	Como p, pero centrado verticalmente [†] .
b{longitud}	Como p, pero verticalmente al fondo [†] .

`table` para tablas que sirve para posicionar y añadir descripciones; funciona exactamente igual.

```
\begin{table}[!h]
  \centering
  \begin{tabular}{|lcc|}
    \hline
      País & Contagios totales & Fallecidos
      totales \\
    \hline \hline
      Argentina & 201,919 & 3,667 \\
      Chile & 361,493 & 9,707 \\
      Colombia & 317,651 & 10,650 \\
      Peru & 428,850 & 19,614 \\
    \hline
  \end{tabular}
  \caption{Estado Covid-19 (3 de agosto de 2020).}
\end{table}
```

País	Contagios totales	Fallecidos totales
Argentina	201,919	3,667
Chile	361,493	9,707
Colombia	317,651	10,650
Peru	428,850	19,614

Cuadro 1: Estado Covid-19 (3 de agosto de 2020).

Se puede agrupar un formato y repetir n veces con la abreviación `*{n}{formato}`. En el ejemplo anterior podríamos reemplazar la `cc` por `*{2}{c}`. Una celda puede usar el tamaño de varias dentro de la misma fila usando el comando `\multicolumn{num}{form}{contenido}` donde num es el número de columnas que abarca, y $form$ el formato de ella, e.g:

```
\begin{table}[!h]
  \centering
```

```

\begin{tabular}{|l|cc|}
\hline
País & Contagios totales & Nuevos contagios \\
\hline \hline
\multicolumn{3}{\sffamily América} \\
\hline
EEUU & 4,851,407 & +38,379 \\
Brasil & 2,736,298 & +2,621 \\
\hline
\multicolumn{3}{\sffamily Europa} \\
\hline
Reino Unido & 305,623 & +928 \\
Italia & 248,229 & +159 \\
\hline
\end{tabular}
\end{table}

```

País	Contagios totales	Nuevos contagios
América		
EEUU	4,851,407	+38,379
Brasil	2,736,298	+2,621
Europa		
Reino Unido	305,623	+928
Italia	248,229	+159

Con el paquete `multirow` puedes combinar filas también mediante el comando `\multirow{num}{longitud}{contenido}` (puedes usar `*` en el lugar de la longitud para ocupar todo lo disponible). Y puedes hacer líneas horizontales parciales desde el inicio de la i -ésima celda hasta el final de la j -ésima celda con `\cline{i-j}`, e.g.:

```

\begin{table}[!h]
\centering
\begin{tabular}{|l|lcc|}
\cline{2-4}
\multicolumn{1}{c|}{} & País & Contagios \\
& totales & Nuevos contagios \\
\hline
\multirow{2}{*}{América} & EEUU & 4,851,407 \\
& & +38,379 \\
\cline{2-4}
{} & Brasil & 2,736,298 \\
& & +2,621 \\
\hline
\end{tabular}
\end{table}

```

```
\end{table}
```

	País	Contagios totales	Nuevos contagios
América	EEUU	4,851,407	+38,379
	Brasil	2,736,298	+2,621

Tablas entre páginas. Por defecto \LaTeX no rompe una tabla entre páginas, sin importar cuan larga sea, para poder usar un *tabla larga*

1.4. Bibliografías

\LaTeX es bastante popular por su forma de tratar bibliografías, por lo cual vamos a enseñar como: En primer lugar, todas las referencias bibliográficas se guardan en otro archivo externo de extensión `.bib`. Para incluir una bibliografía debemos añadir el siguiente comando:

```
\usepackage[backend=biber]{biblatex}
\addbibresource{mis-referencias.bib}
```

Y para compilarlo hay que compilar el documento una primera vez, luego compilar la bibliografía con *biber* y luego una segunda vez. Esto se debe a que la primera vez le dice a \LaTeX que referencias crear, luego el compilador las crea con su respectivo formato y la última las incluye.

Las referencias se pueden buscar facilmente en páginas como <https://scholar.google.com/> y luego seleccionando obtener el formato en Bib \TeX o similar. En general toda referencia suele verse así:

```
@book{gauss1966disquisitiones,
  title={Disquisitiones arithmeticae},
  author={Gauss, Carl Friedrich},
  volume={157},
  year={1966},
  publisher={Yale University Press}
}
```

Donde es bastante claro como funciona, la palabra `book` indica que la referencia es un libro, `gauss1966disquisitiones` es lo que le decimos una *etiqueta*, i.e., una cadena de signos sin espacios que se utiliza para citar dentro del documento, luego el resto es claro. Lo único importante a saber es que en el campo del autor se recomienda anotar apellido, luego coma, luego nombre; y si hay más de uno separarlos por la palabra `and`, si hay muchos y se quiere utilizar algo para decir “y otros” (et al. en latín) se utiliza un

`others` (Albert Einstein et al. sería `Einstein, Albert and others` para \LaTeX).

Usualmente se utiliza un solo archivo general para tener todas las referencias allí, luego, utilizando el ejemplo anterior, uno citaría un documento así:

```
\cite{gauss1966disquisitiones}
```

Si se quiere que la cita este al pie de página uno ocupa `\footcite{...}` y puedes citar varios documentos juntos separando las etiquetas por comas. Entre las referencias sólo aparecerán los trabajos citados. Para incluir un trabajo sin citarlo en la bibliografía debemos usar el comando `\nocite{...}` donde las mismas reglas se aplican. Si se quiere incluir a todas las referencias puedes escribir `\nocite{*}`. Para imprimir las bibliografías se ocupa este comando:

```
\printbibliography
```

Sub-bibliografías y palabras clave. No obstante, un problema que puede surgir al tratar bibliografías es que toman un capítulo (o sección si de un artículo se trata) entero cuando se imprimen, y un formato interesante es el de ciertos libros que usan bibliografía para cada capítulo, lo que por el momento parece una tarea imposible. Un problema similar es si queremos incluir muchos artículos a la vez, pero no todos y sólo queremos incluir los de un cierto tema. Para ambos la solución implica introducir el mismo concepto: palabras clave.

Dentro de la definición de libros en BibTeX se añade la categoría `keywords` (sí, plural) donde especifica las palabras clave separadas por comas. Tomemos el ejemplo del libro de Gauss y supongamos que le añadimos `keywords={arithmetic}` donde tenemos varias referencias marcadas sobre el tema de `arithmetic` luego este comando:

```
\printbibliography[keyword={arithmetic}]
```

Imprimirá sólo dichos artículos. Para el problema de las sub-bibliografías añade la opción de `subbibintoc`. Y para cambiarle el nombre puede usar la opción, `heading={mi título}`.

1.5. Configurar mi documento

Esta sección es opcional, pero se recomienda bastante para mejorar su experiencia y eficiencia con \LaTeX :

(Re) definir comandos. Además de los comandos actuales, se pueden definir nuevos con `\newcommand`, este va seguido del nombre del comando y luego del uso, e.g:

```
\newcommand{\licencia}{Éste documento está bajo la
  licencia xyz (2020).}
-----
\licencia
```

Éste documento está bajo la licencia xyz (2020).

Si quiere añadirle opciones al comando entonces debe poner un campo opcional con el número de opciones y luego referirse a ellas dentro de la definición como `#1`, `#2` y así:

```
\newcommand{\licencia}[2]{Éste documento está bajo la
  licencia #2 (#1).}
-----
\licencia{2019}{uvw}
```

Éste documento está bajo la licencia uvw (2019).

Además también puedes pasar argumentos opcionales, que toman el lugar del primer argumento si están dados y cuyo valor por defecto queda definido en el segundo campo opcional:

```
\newcommand{\licencia}[2][2020]{Éste documento está bajo
  la licencia #2 (#1).}
-----
\licencia{abc}
\licencia[1995]{def}
```

Éste documento está bajo la licencia abc (2020).

Éste documento está bajo la licencia def (1995).

Si un comando ya existe, `LATEX` tirará un error al tratar de sobrescribirlo, para poder hacerlo debes usar `\renewcommand`, que funciona exactamente igual. Éste último también tira error si se trata de redefinir un comando que no ha sido definido antes.

Definir entornos. Si se quiere definir entornos, se ocupa `\newenvironment` y `\renewenvironment` que funcionan exactamente igual, sólo que en lugar de poner un comando pones el tag del entorno y al final se ocupan dos campos obligatorios que determinan el código a ejecutar antes y después del contenido del entorno. Pero todo uso de argumentos extra van en la primera parte y no pueden ir al final, e.g:

```

\newenvironment{demo}[1][\bullet]{#1 {\scshape
  Demostración:} }{\par\noindent}
-----
\begin{demo}
  Esto se deduce de que el triángulo sea isóceles.
\end{demo}
\begin{demo}[(?)]
  Esto es obvio.
\end{demo}

```

- DEMOSTRACIÓN: Esto se deduce de que el triángulo sea isóceles.
(?) DEMOSTRACIÓN: Esto es obvio.

Plantillas. Si usted ha seguido el texto hasta ahora, probando de todo lo que se menciona, es probable que la cabecera de su documento tenga tantas líneas y genere un poco de confusión, sin contar con el hecho de que si piensa hacer otro documento tendrá que copiar y pegar todo eso. Para esto existen las llamadas *plantillas* (*template* en inglés). La manera más fácil es hacer un documento general (usualmente de nombre `template.tex`) en donde guardar todos los comandos y luego importarlo con:

```
\input{template.tex}
```

2. Escribir matemáticas

2.1. Básico

\LaTeX admite dos formas de denotar matemáticas, en modo *entre líneas* y modo *display*. El primero se escribe así `\$1+1=2\$` $1 + 1 = 2$, mientras que el segundo se escribe así:

```
$$1+1=2$$
```

$$1 + 1 = 2$$

Esta distinción es importante, pues \LaTeX tomará ciertas decisiones para no interferir con el resto del texto, lo que hará que el modo entre líneas tenga ciertas peculiaridades. Notemos que por defecto \LaTeX entiende que ha de añadir espacios entre las operaciones, además los números no se inclinan mientras que las letras sí, e.g. `\$a\cdot a=a^2\$` $a \cdot a = a^2$.

\LaTeX también admite subíndices con `_`, e.g., `\$x_1 \leq x_2\$` $x_1 \leq x_2$, pero hemos de tener cuidado con los índices y las potencias pues `\$a^{-1}` da a^{-1} , esto se debe a que \LaTeX sólo interpreta el primer símbolo arriba,

algo similar ocurre con los índices; para usar más de uno juntos se pueden agrupar con {}, e.g. π^{-1} .

Igual que con π , L^AT_EX incluye comandos para varios otros símbolos usuales:

Griego			
$\backslash\alpha$ A\$	αA	$\backslash\nu$ N\$	νN
$\backslash\beta$ B\$	βB	$\backslash\chi\backslash Xi$	$\xi\Xi$
$\backslash\gamma$ \Gamma\$	$\gamma\Gamma$	$\backslash o$ O\$	oO
$\backslash\delta$ \Delta\$	$\delta\Delta$	$\backslash\pi$ \Pi\$	$\pi\Pi$
$\backslash\epsilon$ \varepsilon E\$	$\epsilon\varepsilon E$	$\backslash\rho\backslash varrho$ P\$	$\rho\rho P$
$\backslash\zeta$ Z\$	ζZ	$\backslash\sigma$ \Sigma\$	$\sigma\Sigma$
$\backslash\eta$ H\$	ηH	$\backslash\tau$ T\$	τT
$\backslash\theta$ \vartheta \Theta\$	$\theta\vartheta\Theta$	$\backslash\upsilon$ \Upsilon\$	$\upsilon\Upsilon$
$\backslash\iota$ I\$	ιI	$\backslash\phi$ \varphi \Phi\$	$\phi\varphi\Phi$
$\backslash\kappa$ \varkappa K\$	$\kappa\varkappa K$	$\backslash\chi$ X\$	χX
$\backslash\lambda$ \Lambda\$	$\lambda\Lambda$	$\backslash\psi$ \Psi\$	$\psi\Psi$
$\backslash\mu$ M\$	μM	$\backslash\omega$ \Omega\$	$\omega\Omega$
Operaciones			
$\backslash\cdot$	\cdot	$\backslash\times$	\times
$\backslash\pm$	\pm	\backslashmp	\mp
$\backslash\div$	\div	\backslashcap	\cap
\backslashcup	\cup	\backslashneq	\neq
\backslashleq	\leq	\backslashgeq	\geq
\backslashin	\in	\backslashperp	\perp
\backslashmid	$ $	\backslashparallel	\parallel
\backslashnotin	\notin	\backslashsubset	\subset
\backslashsubteq	\subseteq	\backslashsupseteq	\supseteq
\backslashsimeq	\simeq	\backslashapprox	\approx
\backslashwedge	\wedge	\backslashvee	\vee
\backslashoplus	\oplus	\backslashotimes	\otimes
\backslashcirc	\circ	\backslashsim	\sim
\backslashequiv	\equiv	\backslashcong	\cong
\backslashto	\rightarrow	\backslashimplies	\Rightarrow
Delimitadores			
$\backslash $	$ $	$\backslash $	\parallel
$\backslash\{\}$	$\{ \}$	\backslashlfloor \rfloor\$	$\lfloor \rfloor$
\backslashlceil \rceil\$	$\lceil \rceil$	\backslashlangle \rangle\$	$\langle \rangle$
\backslashulcorner \urcorner\$	$\ulcorner \urcorner$	\backslashllcorner \lrcorner\$	$\llcorner \lrcorner$
Otros			

<code>\infty</code>	∞	<code>\forall</code>	\forall
<code>\Re</code>	\Re	<code>\Im</code>	\Im
<code>\nabla</code>	∇	<code>\exists</code>	\exists
<code>\partial</code>	∂	<code>\nexists</code>	\nexists
<code>\emptyset</code>	\emptyset	<code>\varnothing</code>	\varnothing
<code>\wp</code>	\wp	<code>\complement</code>	\complement
<code>\neg</code>	\neg	<code>\aleph</code>	\aleph
<code>\beth</code>	\beth	<code>\gimel</code>	\gimel
<code>\square</code>	\square	<code>\surd</code>	\surd
<code>\blacksquare</code>	\blacksquare	<code>\triangle</code>	\triangle

Para los marcados con violeta, se requieren dos nuevos paquetes:

```
\usepackage{amsmath, amssymb}
```

Nótese que al estar separados por coma, \LaTeX entiende que son dos paquetes independientes. Si no le interesan dichos símbolos aun se recomienda importar el primero.

Encontrar símbolos. En general es una tarea practicamente imposible la de aprenderse de memoria todos los símbolos en \LaTeX , sin considerar el hecho de que ciertos paquetes importan cada vez más de ellos, por lo que se recomienda con creces usar este sitio web <http://detexify.kirelabs.org/classify.html> donde puedes encontrar símbolos mediante dibujos. Bajo el comando donde aparecen se especifica si el comando funciona en modo texto o modo matemático (o ambos), y arriba especificará si requiere algún paquete.

Otra observación es que si bien `|` sirve y genera el mismo efecto en modo matemático que `\mid`, el primero se recomienda como *envoltura* (similar a los paréntesis), mientras que el segundo genera espacios como para el “divide a”, e.g., `\mid x|`; `2\mid 4` $|x|$; $2 \mid 4$. Esta distinción aplica para el resto de delimitadores.

Puntos. Pese a que no sea obvio, en \LaTeX uno posee varios tipos de puntos: centrados, bajos, verticales y diagonales (del tipo \ddots). Para ello se escribe el punto centrado solo como `\cdot` como hemos visto, pero los tres juntos se escriben `\cdots`, `\dots`, `\vdots`, `\ddots` \cdots , \dots , \vdots , \ddots respectivamente.

Como recomendación se usa los puntos centrados para operaciones, e.g., `1+2+\cdots+99+100` $1 + 2 + \cdots + 99 + 100$, y los bajos para listas `\{1,2,\dots,100\}` $\{1, 2, \dots, 100\}$.

Fracciones. Para hacer fracciones existe el comando `\frac` que funciona como:

```
$$ \frac{1}{2} + \frac{1}{3} = \frac{5}{6}. $$
```

$$\frac{1}{2} + \frac{1}{3} = \frac{5}{6}.$$

No obstante, el comando cambia si estamos en modo texto: `\frac{22}{7}` y también lo hace si ponemos varias fracciones juntas:

```
$$ \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{\ddots} $$$
```

$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \ddots}}}$$

Esto se debe a que en el numerador y denominador se pasa a modo entre líneas para no ocupar demasiado espacio, en el modo entre líneas se utiliza para que las líneas se mantengan constantes en altura, no obstante, de ser necesario puede usar `\dfrac` para forzar una fracción en modo *display* y `\tfrac` para forzar modo entre líneas.

```
$$ \dfrac{1}{1} + \dfrac{1}{1} + \dfrac{1}{1} + \dfrac{1}{\ddots} $$$
```

$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \ddots}}}$$

```
$$ \frac{2}{5} + \frac{39}{45} = \frac{57}{45} = 1\tfrac{12}{45} $$$
```

$$\frac{2}{5} + \frac{39}{45} = \frac{57}{45} = 1\frac{12}{45}$$

Un comando similar es `\binom` que genera coeficientes binomiales, e.g., `\binom{3}{2}`.

Raíces. Este es muy sencillo, así se describe una raíz cuadrada `\sqrt{17}`, y así una raíz n -ésima `\sqrt[3]{8}` = `2\sqrt[3]{8}` = 2.

Operadores. En general a los símbolos destinados a ir entre dos caracteres como `+` se les dice “operación binaria”, por separado, un operador es una palabra o frase que se utiliza explícitamente y, por lo general, en letras romanas derechas; e.g., `\sin x` sin x .

Los operadores se clasifican en dos: grandes o pequeños, donde ambos ocupan el mismo tamaño para su nombre, pero los grandes tienen la gracia de que sus subíndices y superíndices van directamente abajo y arriba respectivamente del nombre, por ejemplo:

```
$$ \lim_{x \to \infty} \sin^2 x + \cos^2 x = 1. $$
```

$$\lim_{x \rightarrow \infty} \sin^2 x + \cos^2 x = 1.$$

Aquí `\sin`, `\cos` son pequeños mientras que `\lim` es grande.

Los operadores no siempre tienen que tener texto, pueden poseer otros símbolos como:

```
$$ \sum_{i=0}^n i = \frac{n(n+1)}{2}, \int_a^b f(x) dx. $$
```

$$\sum_{i=0}^n i = \frac{n(n+1)}{2}, \int_a^b f(x) dx.$$

Cabe notar que dicha característica de los operadores grande se pierde en el modo entre líneas, e.g., `\prod_{k=1}^n i = n!`

<code>\cos</code>	cos	<code>\csc</code>	csc
<code>\exp</code>	exp	<code>\ker</code>	ker
<code>\sinh</code>	sinh	<code>\arcsin</code>	arcsin
<code>\cosh</code>	cosh	<code>\deg</code>	deg
<code>\lg</code>	lg	<code>\ln</code>	ln
<code>\arctan</code>	arctan	<code>\cot</code>	cot
<code>\hom</code>	hom	<code>\log</code>	log
<code>\sec</code>	sec	<code>\tan</code>	tan
<code>\arg</code>	arg	<code>\coth</code>	coth
<code>\dim</code>	dim	<code>\sin</code>	sin
<code>\tanh</code>	tanh		
<code>\liminf</code>	lím inf	<code>\limsup</code>	lím sup
<code>\inf</code>	ínf	<code>\sup</code>	sup
<code>\min</code>	mín	<code>\max</code>	máx
<code>\lim</code>	lím	<code>\gcd</code>	gcd
<code>\Pr</code>	Pr	<code>\det</code>	det
<code>\int</code>	\int	<code>\iint</code>	\iint
<code>\iiint</code>	\iiint	<code>\idotsint</code>	$\int \cdots \int$
<code>\oint</code>	\oint	<code>\coprod</code>	\coprod
<code>\sum</code>	\sum	<code>\prod</code>	\prod
<code>\bigcup</code>	\bigcup	<code>\bigcap</code>	\bigcap
<code>\bigvee</code>	\bigvee	<code>\bigwedge</code>	\bigwedge

En la lista los primeros son los pequeños y los segundos los grandes.

Definir operadores. Para definir operaciones se debe utilizar el comando `\DeclareMathOperator{com}{tag}` en la cabecera. Si se declara con un `*` antes de las llaves, entonces se definirá un operador grande, e.g:

```
\DeclareMathOperator{\mcd}{mcd}
\DeclareMathOperator*{\dom}{dom}
-----
$$ \mcd(21, 70) = 7, \quad \dom_V(f) = A. $$
```

$$\text{mcd}(21, 70) = 7, \quad \text{dom}_V(f) = A.$$

Pero esto es una definición general, aquí defines un comando para todo el documento lo que puede o no ser de su agrado, si tiene un operador que desea ocupar una única vez entonces es más recomendable usar `\operatorname`; sus efectos son los mismos:

```
$$ \operatorname{mcm}(3, 16) = 48, \quad \operatorname{cod}_V(f) = B. $$
```

$$\text{mcm}(3, 16) = 48, \quad \text{cod}_V(f) = B.$$

También podemos utilizar el anterior para redefinir comandos, por ejemplo si no nos gusta que `\Re`, `\Im` se vean así \Re , \Im , podemos poner esto en la cabecera:

```
\renewcommand{\Re}{\operatorname{Re}}
\renewcommand{\Im}{\operatorname{Im}}
```

Para que se vean así Re , Im .

Espacios. El ejemplo anterior también sirve para ilustrar una característica del modo matemático, la cual es que no lee los espacios singulares del código, e.g., `$a b$` genera ab . Para añadir espacios también hay comandos singulares:

<code>\$a\!b\$</code>	ab
<code>\$ab\$</code>	ab
<code>\$a\,b\$</code>	$a\,b$
<code>\$a\;b\$</code>	$a\;b$
<code>\$a\ b\$</code>	$a\ b$
<code>\$a\quad b\$</code>	$a\quad b$
<code>\$a\qquad b\$</code>	$a\qquad b$

Fuentes de letras. Para el modo matemático también hay fuentes de letras:

<code>\mathrm</code>	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
<code>\mathbf</code>	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
<code>\mathsf</code>	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
<code>\mathbb</code>	ABCDEFGHIJKLMNOPQRSTUVWXYZ
<code>\mathcal</code>	<i>ABCDEFGHIJKLMNOPQRSTUVWXYZ</i>
<code>\mathfrak</code>	<i>ABCDEFGHIJKLMNOPQRSTUVWXYZ</i> <i>abcdefghijklmnopqrstuvwxyz 1234567890</i>
<code>\mathscr</code>	<i>A B C D E F G H I J K L M</i> <i>N O P Q R S T U V W X Y Z</i>

Es común usar `\mathbb` para conjuntos numéricos famosos como \mathbb{N} . A veces, éstos suelen ser tan comunes y conocidos, que suele ser útil definir un comando para ellos:

```
\newcommand{\N}{\mathbb{N}}
\newcommand{\Z}{\mathbb{Z}}
\newcommand{\Q}{\mathbb{Q}}
\newcommand{\R}{\mathbb{R}}
\newcommand{\C}{\mathbb{C}}
```

Texto en ecuaciones. Si bien la fuente `\mathrm` es la misma del texto del documento, todavía tendríamos que lidiar con los espacios del modo matemático sin hablar de que no admiten caracteres `utf-8` (i.e., cosas como $\$ \text{á} \$$ generan errores), pero L^AT_EX admite texto en ecuaciones, sólo basa usar `\text`, e.g. $\$x=2, \text{ mientras que } y=3\$$ $x = 2$, mientras que $y = 3$. Esto es más útil en ecuaciones en modo *display*.

2.2. Ecuaciones numeradas, alineadas, matrices y más

Ecuaciones numeradas. Hasta ahora nuestras ecuaciones no están numeradas, ni poseen un número para añadirlas, para hacerlo debes usar el entorno `equation`. Dentro de él se permiten el uso de referencias cruzadas, respecto a esto se recomienda usar `\eqref` en lugar de `\ref` por estética, aunque sus usos son iguales, e.g:

```
...y así se deduce el llamado \textit{teorema de Pitágoras}
\eqref{eq:pitagoras}:
\begin{equation} \label{eq:pitagoras}
```

```
a^2 + b^2 = c^2
\end{equation}
```

...y así se deduce el llamado *teorema de Pitágoras* (1):

$$a^2 + b^2 = c^2 \quad (1)$$

Como se ve, L^AT_EX automáticamente enumera la ecuación como 1, si se quiere cambiar dicho tag se puede mediante el comando `\tag`, e.g:

```
\begin{equation}
    E = mc^2 \label{eq:einstein} \tag{\heartsuit}
\end{equation}
Y \eqref{eq:einstein} es la famosa fórmula de Einstein...
```

$$E = mc^2 \quad (\heartsuit)$$

Y (♥) es la famosa fórmula de Einstein...

Además se puede quitar los tags con `\notag`, que tal vez no es útil aún, pero lo será pronto.

Ecuaciones alineadas. Es común ver documentos donde una ecuación está alineada por signos “=” y no saber como, además notará que entre `$$` ... `$$` no funcionan los cortes de línea. Para ello existe el entorno `align` y `align*`. Donde las partes son como celdas de una tabla y se separan por `&`. El primero enumera cada línea, mientras que el segundo no enumera ninguna, e.g:

```
\begin{align}
    \sum_{i=1}^n i &= 1 + 2 + 3 + \cdots + (n-1) + n \notag \\
    &= (1 + n) + (2 + (n-1)) + (3 + (n-2)) + \cdots \notag \\
    &= (n+1) + (n+1) + (n+1) + \cdots \notag \\
    &= (n+1) \cdot \frac{n}{2} = \frac{n(n+1)}{2}.
\end{align}
```

$$\begin{aligned}
 \sum_{i=1}^n i &= 1 + 2 + 3 + \cdots + (n-1) + n \\
 &= (1 + n) + (2 + (n-1)) + (3 + (n-2)) + \cdots \\
 &= (n+1) + (n+1) + (n+1) + \cdots \\
 &= (n+1) \cdot \frac{n}{2} = \frac{n(n+1)}{2}.
 \end{aligned} \quad (2)$$

Un detalle importante es que por defecto L^AT_EX no rompe una ecuación alineada entre páginas, sin importar cuan larga sea, si usted quiere permitir esto debe agregar este comando a la cabecera:

```
\allowdisplaybreaks
```

Ecuación multi-línea. Ésta es diferente de la anterior, en esta se usan varias líneas, pero todos bajo un mismo número, o varias ecuaciones, pero sin alinear. Para una ecuación muy larga se puede usar el entorno `multline` y para varias ecuaciones no-alineadas el entorno `gather`. El asterisco indica si van enumeradas o no:

```
\begin{multline*}
f(x) = 59x^{11} + 86x^{10} + 22x^9 + 83x^8 + 74x
      ^7 + 49x^6 + 12x^5 + 83x^4 \\
      + 42x^3 + 69x^2 + 47x + 93.
\end{multline*}
y
\begin{gather*}
g(x) = 47x^2 + 62x + 45 \\
h(x) = 18x^5 + 21x^3 + 61x - 9
\end{gather*}
```

$$f(x) = 59x^{11} + 86x^{10} + 22x^9 + 83x^8 + 74x^7 + 49x^6 + 12x^5 + 83x^4 + 42x^3 + 69x^2 + 47x + 93.$$

y

$$g(x) = 47x^2 + 62x + 45$$
$$h(x) = 18x^5 + 21x^3 + 61x - 9$$

Modo entre-línea, display y smash. A mucha gente le molesta el hecho de que \LaTeX achica ciertos símbolos para mantener constancia en sus líneas, en cosas como $\sum_{k=1}^n k$ y otros operadores grandes. Ya vimos el problema con `\dfrac`, pero no vimos el caso general para `\lim` y el resto de operadores grandes. Para esto \LaTeX posee los modificadores `\displaystyle` y `\textstyle` que fuerzan el modo display y entre-líneas respectivamente en cualquier entorno. Personalmente no recomiendo esto por mal uso de líneas, e.g. `\displaystyle \sum_{k=1}^n k` provoca malos espacios; no obstante hay casos donde se podría hacer sin afectar el espacio, para ello está el comando `\smash`:

```
Línea corta. \\
Así que ahora podemos usar \smash{\displaystyle \sum_{k=1}^n k
  $}. \\
Línea corta.
```

Línea corta.

Así que ahora podemos usar $\sum_{k=1}^n k$.

Línea corta.

Delimitadores grandes. Otro problema de tamaño es que por defecto los delimitadores no se adaptan al tamaño de símbolos grandes en L^AT_EX, e.g., $(\frac{1}{2})$ que se ve mal. Para ello se recomienda usar `\left` y `\right`, e.g.

`\left(\dfrac{1}{2}\right)`. Estos se adaptan a cualquier símbolo, sin importar que tan grande, sólo se debe ser cuidadoso, **no se puede** usar `\left` y luego no ocupar un `\right` en la misma línea. En caso de que sólo se quiera usar un delimitador se debe poner un ‘.’ en el otro, e.g:

```
$$ \int_0^2 x^2 dx = \left. \frac{x^3}{3} \right|_0^2 = \frac{8}{3}. $$
```

$$\int_0^2 x^2 dx = \frac{x^3}{3} \Big|_0^2 = \frac{8}{3}.$$

Para cambiar el tamaño de forma manual puede usar los siguientes:

<code>\big(\$</code>	<code>(</code>	<code>\Big(\$</code>	<code>(</code>
<code>\bigg(\$</code>	<code>(</code>	<code>\Bigg(\$</code>	<code>(</code>

Matrices. Para escribir matrices, el paquete `amsmath` incluye el entorno `matrix` para el modo matemático que funciona como una tabla:

```
$$ \mathbf{Id} :=
\left[ \begin{matrix}
1 & 0 \\
0 & 1
\end{matrix} \right] $$
```

$$\mathbf{Id} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Por defecto, el entorno no posee delimitadores, pero existen variaciones de él que sí:

<code>matrix</code>	$\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}$	<code>pmatrix</code>	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
<code>bmatrix</code>	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	<code>Bmatrix</code>	$\left\{ \begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix} \right\}$
<code>vmatrix</code>	$\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$	<code>Vmatrix</code>	$\left\ \begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix} \right\ $