

Una guía de L^AT_EX

Alejandro Cholaquidis

Centro de Matemática
Facultad de Ciencias
Universidad de la República

Índice general

1. Introducción	4
1.1. Estructura de los documentos	4
1.1.1. Documentos grandes	6
1.2. Ingreso de texto	6
1.2.1. Caracteres especiales	6
1.2.2. Espaciado	6
1.2.3. Cortar palabras	7
1.2.4. Alineación de Texto	8
1.3. Ingresar fórmulas matemáticas	8
1.4. Referencias	8
1.5. Pie de Página	9
2. Formato del texto	10
2.1. Cursiva, subrayado, etc	10
2.2. Listas	10
2.2.1. Enumeraciones	10
2.2.2. Itemize	11
2.3. Tablas	11
3. Fórmulas y Símbolos Matemáticos	13
3.1. Ingreso de texto	13
3.1.1. Espacio entre símbolos	13
3.1.2. Tipos de letra	13
3.1.3. Acentos	14
3.1.4. Operadores	14
3.1.5. Paréntesis	14
3.1.6. Fracciones y fracciones anidadas	15
3.1.7. Alineación de símbolos	15
3.2. Entornos matemáticos	15
3.3. Matrices	16
3.3.1. Entorno array	16
3.3.2. Otros entornos	17
3.4. Teoremas, Lemas, etc	17
3.4.1. Teoremas y demostraciones	17
3.4.2. Lemas	18
3.4.3. Corolarios, Proposiciones, etc	18
4. Encabezados y márgenes	19
4.1. Márgenes	19
4.2. Encabezado y pie de página	20

4.2.1. Encabezado	20
4.2.2. Pie de Página	20
4.2.3. Encabezados distintos para páginas pares e impares	20
5. Diagramas	22
5.1. Diagramas de una fila	22
5.1.1. Variantes de flechas	23
5.2. Diagramas de 2 o más filas	23
5.2.1. Otras flechas	24
6. Otros tópicos	25
6.1. Objetos Flotantes	25
6.2. Insertar imágenes	25
6.3. Índices	26
6.4. Bibliografía y apéndices	26
6.4.1. Bibliografía	26
6.4.2. Apéndice	26
6.5. Crear comandos	27
<i>Bibliografía</i>	28

Capítulo 1

Introducción

Antes que nada aclaramos que estas páginas sólo pretenden ser una guía inicial al \LaTeX , sin perjuicio de que confiamos en que pueda ser útil. Un segundo paso, o simultáneo, puede ser consultar [3], y luego [2]. Ya [1] es una obra mucho más completa.

En este primer capítulo explicaremos los comandos básicos necesarios para escribir un documento sencillo en \LaTeX e ingresar algunas fórmulas matemáticas básicas. A veces se mostrará a la izquierda del documento el código tal cual se ingresa en \LaTeX y a la derecha el resultado que se observa una vez generado el documento.

1.1. Estructura de los documentos

Todo documento de \LaTeX debe comenzar con el comando:

```
\documentclass[opciones]{clase}.
```

En *clase* se indica el tipo de documento que queremos crear. Dentro de las posibles clases tenemos: *article*, *report*, *book*, *letter*.

article

Se emplea en general para escribir, como su nombre lo indica pequeños artículos. Sus posibles subdivisiones son:

```
\part{}, \section{}, \subsection{}, \subsubsection{}, \paragraph{}, \subparagraph{}
```

report

A diferencia de *article*, *report* permite, como división intermedia entre `\part{}` y `\section{}` la división en capítulos mediante `\chapter{}`.

book

Permite las mismas subdivisiones que *report* pero a diferencia de éste y *article*, se toma por defecto que la escritura es en doble faz, se agrega además un encabezado en cada hoja, donde en las pares se indica el nombre del capítulo y en las impares el de la sección.

letter

Tiene divisiones específicas para la creación de cartas:

```
\signature{}, \address{}, \opening{}, \closing{}
```

En opciones indicamos opciones específicas para la clase elegida, las diferentes opciones se separan con una coma, tenemos:

Tamaño de Letra

Por ejemplo: `10pt, 11pt, 12pt`, por defecto si no se indica nada se asume `10pt`.

Tamaño de Hoja

`a4paper`, `letterpaper` (opción por defecto), `a5paper`, `b5paper`, `executivepaper`.

Orientación de las Ecuaciones

Si bien por defecto las ecuaciones se centran, con la opción `fleqn` las orientamos hacia la izquierda.

Numeración de Ecuaciones

Por defecto las ecuaciones se numeran a la derecha, con `leqno` las numeramos a la izquierda.

Columnas

Si bien por defecto se escribe en una columna, con `twocolumn` podemos escribir en dos.

Doble faz

Con `twoside` y `oneside` podemos elegir si escribimos en doble faz o no, por defecto `article` y `report` son `oneside` mientras que `book` es `twoside`.

Comienzo de capítulo

Con la opción `openright` le indicamos a latex que los capítulos comienzan solamente en páginas impares.

Luego de que hemos especificado en la primera línea el tipo de documento que vamos a escribir, antes de comenzar el documento, debemos indicarle a \LaTeX los paquetes que vamos a usar. Un paquete puede pensarse como un conjunto de comandos que nos permitirán ingresar caracteres especiales así como estructuras particulares que no se encuentran por defecto en \LaTeX . Cada conjunto de comandos se debe habilitar antes, mediante la opción:

```
\usepackage[opción]{nombre del paquete}
```

La siguiente lista contiene algunos de los paquetes más usados:

`inputenc` En opciones le indicamos la codificación que vamos a usar; por ejemplo, si queremos que funcionen los tildes en \LaTeX tenemos que poner `\usepackage[utf8x]{inputenc}`.

`babel` En opciones le indicamos el idioma. Por ejemplo: `\usepackage[spanish]{babel}`

\LaTeX Paquetes de símbolos matemáticos y tipos de letra creados por la AMS (American Mathematical Society). Entre ellos tenemos: `amsfonts`, `amssymb`, `amsmath`, `amsthm`.

`graphicx` Para ingresar gráficos.

`fancyhdr` Configurar lo referente a encabezados, pie de páginas y márgenes.

`Xy` Paquete para ingresar diagramas.

`color` Para cambiar el color de la letra.

array Paquete que permite poner distintas clases de matrices; se verá más adelante.

multirow Paquete para combinar columnas y filas en un array.

Luego de definido el tipo de documento y cargados los paquetes necesarios, estamos en condiciones de crear nuestro pequeño documento. Más adelante veremos cómo configurar el tamaño de los márgenes, poner encabezados, etc, lo cual se hace antes de empezar el documento, en el área a la cual nos referiremos como preámbulo.

Todo documento se escribe entre los comandos:

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

1.1.1. Documentos grandes

A veces queremos que capítulos distintos estén en archivos distintos, por ejemplo si el documento que vamos a escribir es muy grande, pues esto nos permite ordenar mejor el texto y buscar mejor dentro de él. Lo que tenemos que hacer entonces es, entre el `\begin{document}` y el `\end{document}` del documento principal, es decir, el que compilamos y que contiene `\documentclass`, agregar las líneas `\include{nombre}`, donde nombre refiere al nombre del archivo que vamos a incluir. Por ejemplo:

```
\begin{document}
\include{capitulo1}
\include{capitulo2}
\include{bibliografia}
\end{document}
```

Cada capítulo aparte que se crea se comienza con `\chapter{}`. Si lo que queremos es adjuntar una parte que contenga varios capítulos, se pone el correspondiente `\include{parte1}` y se comienza ese archivo con `\part{}`.

1.2. Ingreso de texto

En esta sección vamos a ver lo referente al ingreso de texto y de fórmulas matemáticas.

1.2.1. Caracteres especiales

Existen ciertos caracteres que \LaTeX se reserva, y por lo tanto al ingresarlos no *aparecen* si no es indicándoselo a \LaTeX de alguna forma.

Veamos cuáles son y cómo se ingresan:

```
\# \$ \% \^{} \& \_ \{ \} \~{}
\textbackslash
```

\$ % ^ & _ { } ~ \

1.2.2. Espaciado

Sangrías

Si al ingresar texto se baja una línea esto produce que el texto en la siguiente línea comience con una sangría. Para forzar a \LaTeX a sacar ese espacio tenemos que poner `\noindent` antes del

párrafo. Si queremos modificar el espacio que se deja por defecto al comienzo de cada párrafo (o sea: la sangría), debemos agregar antes de `\begin{document}` (por ejemplo para que sea de 2cm):

`\setlength{\parindent}{2cm}`. Podemos hacer que el efecto del cambio de sangría no sea global, agregando el comando antes del `\indent` que queremos agregar, por ejemplo:

`{\setlength{\parindent}{6cm} \indent}`.

Una forma de bajar una línea y no dejar sangría es usando `\\`.

Espacio entre palabras

El efecto logrado al ingresar espacio entre palabras es siempre el mismo, independiente de la cantidad de espacio en cuestión: el resultado es que las palabras estarán separadas por un único espacio. Por ejemplo, las siguientes líneas producen el mismo resultado:

Ejemplo ilustrativo

Ejemplo ilustrativo

Para ingresar espacio entre palabras tenemos que agregar en el texto `\` entre las palabras, cada `\` agrega un espacio. Es decir, para el primer ejemplo:

Ejemplo `\\ \\` ilustrativo

Ejemplo ilustrativo

Otra forma es usando el comando `\hspace{}`:

Ejemplo `\hspace{2cm}` ilustrativo

Ejemplo ilustrativo
--

Si queremos agregar un espacio vertical de por ejemplo 2cm usamos `\vspace{2cm}` o `\\[2cm]`. Para agregar una línea nueva escribimos `\newline` o una página nueva con `\newpage`. Si lo que queremos es cambiar el interlineado de *todo* el texto agregamos antes de `\begin{document}` el comando `\renewcommand{\baselinestretch}{Número}`. Si solamente queremos cambiar el interlineado de un párrafo dado, agregamos el paquete `setspace`, y encerramos el texto entre `\begin{spacing}{nro}` y `\end{spacing}`. El texto de este párrafo tiene un interlineado de 1.5.

1.2.3. Cortar palabras

Observemos que por defecto al terminar una línea \LaTeX separa las palabras en sílabas. En caso de que no queramos usar la separación que usa para una determinada palabra, la separamos manualmente escribiendo dando opciones a \LaTeX acerca de donde separar. Por ejemplo, a continuación se ofrecen dos opciones, de las cuales \LaTeX eligió la primera:

El problema se resolvió

`anti\`-constitucional`\`-mente

El problema se resolvió anti-constitucionalmente
--

1.2.4. Alineación de Texto

Para escribir texto alineado a la izquierda, lo escribimos entre `\begin{flushleft}` y `\end{flushleft}`. Para alinearlo a la derecha lo escribimos entre `\begin{flushright}` y `\end{flushright}`, y para centrarlo entre `\begin{center}` y `\end{center}`.

texto a izquierda

texto centrado

texto a derecha

1.3. Ingresar fórmulas matemáticas

Las fórmulas matemáticas se ingresan entre `$ $` dentro de un párrafo. Si se ponen entre `$$` y `$$` bajan una línea y se centran. Por ejemplo `$$\sum_{i=1}^n x^n` da como resultado $\sum_{i=1}^n x^i$ en el mismo párrafo, mientras que `$$\sum_{i=1}^n x^i` produce

$$\sum_{i=1}^n x^i$$

Obsérvese que el formato en el que aparece dentro del párrafo es distinto. Para que aparezca con el formato desplegado dentro del párrafo hay que agregar `\displaystyle` antes; por ejemplo

`$$\displaystyle \sum_{i=1}^n x^i` produce $\sum_{i=1}^n x^i$. Si queremos que se genere un número identificador de la ecuación podemos usar el comando `\begin{equation}`.

```
\begin{equation}
\sum_{i=1}^n x^i
\end{equation}
```

$$\sum_{i=1}^n x^i \quad (1.1)$$

Si queremos ingresar texto dentro de una fórmula, podemos usar el comando `\textrm{texto}`. Por ejemplo: `$$\ldots n_0 \textrm{ tal que para todo } n > n_0 \textrm{ existe } \ldots` produce: $\dots n_0$ tal que para todo $n > n_0$ existe...

1.4. Referencias

Supongamos que queremos hacer referencia a una ecuación ingresada con `\begin{equation}` e indicar en qué página se encuentra. Para eso, en el lugar al cual queremos hacer referencia, escribimos el comando `\label{nombre}` y luego nos referimos a él como `\ref{nombre}` en la página `\pageref{nombre}`. En el ejemplo anterior:

```
\begin{equation}
\label{suma}
\sum_{i=1}^n x^i
\end{equation}
```

En la ecuación `\ref{suma}` de la página `\pageref{suma}`.

$$\sum_{i=1}^n x^i \quad (1.2)$$

En la ecuación 1.2 de la página 8.

1.5. Pie de Página

Para agregar una nota al pie de la página escribimos, en el lugar donde queremos hacer la nota, el comando `\footnote{nota al pie}`. Por ejemplo acá¹ ponemos una primera nota al pie, mientras que aquí² ponemos otra.

¹primera nota al pie

²segunda nota al pie

Capítulo 2

Formato del texto

En este capítulo vamos a ver cómo modificar el formato del texto, ingresar una lista con ítems y enumerarlos, crear tablas, etc.

2.1. Cursiva, subrayado, etc

Los siguientes comandos permiten cambiar el formato del texto:

```
\textbf{texto en negrita}  
\textit{texto cursiva}  
\underline{subrayado}  
\small{texto en small}  
\large{texto en large}  
\Large{texto en Large}  
\Huge{texto en huge}  
\textcolor{red}{Rojo}
```

texto en negrita
texto cursiva
subrayado
texto en small
texto en large
texto en Large
texto en huge
Rojo

2.2. Listas

2.2.1. Enumeraciones

Si queremos generar una lista podemos usar el entorno `\begin{enumerate} \end{enumerate}`. Por ejemplo

```
\begin{enumerate}  
\item Primer ítem de la lista  
\begin{enumerate}  
\item Primero de la lista anidada  
\item Segundo de la lista anidada  
\end{enumerate}  
\item Segundo ítem de la lista  
end{enumerate}
```

1. Primer ítem de la lista
 - a) Primero de la lista anidada
 - b) Segundo de la lista anidada
2. Segundo ítem de la lista

Por defecto, los ítems se numeran como en el ejemplo, pero esto puede cambiarse agregando antes de `\begin{document}` alguno de los siguientes comandos:

```
\renewcommand{\labelenumi}{\arabic{enumi}.} 1., 2., 3.,...
```

```
\renewcommand{\labelenumi}{\roman{enumi}.}    i., ii., iii.,...
\renewcommand{\labelenumi}{\Roman{enumi}.}    I., II., III.,...
\renewcommand{\labelenumi}{\alph{enumi}.}    a., b., c.,...
\renewcommand{\labelenumi}{(\alph{enumi}).}  (a), (b), (c),...
```

2.2.2. Itemize

Otro entorno que se puede usar y es análogo al anterior (la diferencia radica en que cada ítem se marca con un símbolo) es `\begin{itemize} \end{itemize}`. En este caso para cambiar el símbolo usamos `\renewcommand{\labelitemi}{carácter}`. Y para cambiar los caracteres del segundo, tercer y cuarto nivel de anidamiento, utilizamos los comandos `\labelitemii`, `\labelitemiii` o `\labelitemiv`, respectivamente.

2.3. Tablas

Para generar una tabla usamos el entorno `\begin{tabular}[pos]{especificaciones} \end{tabular}`, donde con *pos* indicamos la posición de la tabla relativa al texto: `t` para indicar top (arriba), `b` bottom (abajo) y `c` center (centro). En especificaciones indicamos la cantidad de columnas, la posición que tendrá el texto dentro de ellas, y eventualmente el ancho que tendrá. Por ejemplo:

```
\begin{tabular}{c|r|l}
f 1 c 1 & f 1 c 2 & f 1 c 3 \\
f 2 c 1 & f 2 c 2 & f 2 c 3
\end{tabular}
```

f 1 c 1	f 1 c 2	f 1 c 3
f 2 c 1	f 2 c 2	f 2 c 3

Se crea una tabla con 3 columnas, donde en la primera, con la letra `c`, se indica que el texto irá centrado. En la segunda la letra `r` indica que el texto irá alineado a la derecha (de “right”, derecha), y en la tercera a la izquierda, porque hay una letra `l` (de “left”, izquierda). La separación `|` indica que se va a agregar una línea vertical entre las columnas. Si en el ejemplo anterior quisiéramos especificar que la columna del medio tiene un ancho de 5cm deberíamos poner `p{5cm}`. Para poner una línea horizontal usamos el comando `\hline`. Si queremos que vaya solamente desde la columna *i* a la *j* usamos `cline{i-j}`

```
\begin{tabular}{c|r|l}
f 1 c 1 & f 1 c 2 & f 1 c 3 \\
\hline
f 2 c 1 & f 2 c 2 & f 2 c 3
\cline{2-2}
\end{tabular}
```

f 1 c 1	f 1 c 2	f 1 c 3
f 2 c 1	f 2 c 2	f 2 c 3

Supongamos que queremos agregar un título a la tabla, y unir filas. Esto se hace con los comandos `\multicolumn{no de columnas}{alineación}{texto}` y `\multirow{nro de filas}{ancho}{texto}` (si no queremos especificar el ancho ponemos `*`). Veamos un ejemplo:

```
\begin{tabular}{c|r|l}
\multirow{2}{*}{f1 y f2 c1} & f 1 c 2 & f 1 c 3 \\
\cline{2-3}
& f 2 c 2 & f 2 c 3
\end{tabular}
```

```
\hline
      f3 c1          & \multicolumn{2}{c}{2 columnas juntas}\\
\end{tabular}
```

produce

f1 y f2 c1	f 1 c 2	f 1 c 3
	f 2 c 2	f 2 c 3
f 3 c1	2 columnas juntas	

En caso de querer modificar la distancia entre las filas o columnas podemos usar los comandos (por ejemplo para 1.2 cm de interlineado y 0.4 cm de separación entre columnas):

```
\renewcommand{\arraystretch}{1.2}
\renewcommand{\tabcolsep}{0.4cm}
```

Se puede hacer que estos parámetros tengan un efecto local, simplemente poniendolos antes de la tabla que queremos:

```
{\renewcommand{\arraystretch}{2.2}
\renewcommand{\tabcolsep}{0.4cm}
\begin{tabular}{c|c}
  a & b\\
\hline
  c & d
\end{tabular}
}
```

a	b
c	d

Capítulo 3

Fórmulas y Símbolos Matemáticos

En el presente capítulo vamos a ampliar lo que vimos en 1.3.

3.1. Ingreso de texto

3.1.1. Espacio entre símbolos

Al ingresar texto en una fórmula, es decir, entre $\$$, doble $\$$, o un entorno matemático como por ejemplo `\begin{equation}` no se deja espacio entre símbolos, palabras etc. Para generar un espacio o quitarlo podemos usar los siguientes comandos:

```
\sqrt{x} \ + \ 1  
\sqrt{x} \quad + 1  
\sqrt{x} \qquad +1  
\int \int x dx  
\int \! \int x dx  
\int \negmedspace \int x dx  
\int \negthickspace \int x dx  
\iint x dx
```

$\sqrt{x} + 1$
$\sqrt{x} \quad + 1$
$\sqrt{x} \qquad + 1$
$\int \int x dx$
$\int \! \int x dx$
$\int \int x dx$
$\int \int x dx$
$\int \int x dx$
$\int \int x dx$

3.1.2. Tipos de letra

Al igual que como vimos en 2.1, en el modo matemático podemos cambiar la tipografía del texto. Algunas opciones son las siguientes:

normal	<code>\mathbf</code>	<code>\mathsf</code>	<code>\mathit</code>	<code>\mathcal</code>	<code>\mathbb</code>	<code>\mathfrak</code>
X	\mathbf{X}	X	X	\mathcal{X}	\mathbb{X}	\mathfrak{X}
x	\mathbf{x}	x	x	\mathcal{x}	\mathbb{x}	\mathfrak{x}
Σ	$\mathbf{\Sigma}$	$\mathsf{\Sigma}$	$\mathit{\Sigma}$	$\mathcal{\Sigma}$	$\mathbb{\Sigma}$	$\mathfrak{\Sigma}$

Obsérvese que en el modo matemático, `\mathbf` no pone en negrita los símbolos, para eso tenemos que usar el comando `\boldsymbol`.

3.1.3. Acentos

\acute{x} <code>\acute{x}</code>	\bar{x} <code>\bar{x}</code>	\vec{x} <code>\vec{x}</code>	\widetilde{xxx} <code>\widetilde{xxx}</code>
\grave{x} <code>\grave{x}</code>	\breve{x} <code>\breve{x}</code>	\dot{x} <code>\dot{x}</code>	\widehat{xxx} <code>\widehat{xxx}</code>
\ddot{x} <code>\ddot{x}</code>	\check{x} <code>\check{x}</code>	\ddot{x} <code>\ddot{x}</code>	\hat{x} <code>\hat{x}</code>
\tilde{x} <code>\tilde{x}</code>			

3.1.4. Operadores

<code>\arccos</code>	<code>\det</code>	<code>\inf</code>
<code>\arcsin</code>	<code>\dim</code>	<code>\lim</code>
<code>\arctan</code>	<code>\exp</code>	<code>\liminf</code>
<code>\arg</code>	<code>\log</code>	<code>\min</code>
<code>\cos</code>	<code>\ln</code>	
<code>\cosh</code>	<code>\ker</code>	
<code>\cot</code>	<code>\mín</code>	
<code>\coth</code>	<code>\hom</code>	

Si queremos crear nuestro propio operador lo podemos hacer incluyendo en el preámbulo el paquete `amsmath`, y luego usando, también en el preámbulo, el comando:

`\DeclareMathOperator{}{}` por ejemplo:

```
\DeclareMathOperator{\rank}{rank}
\DeclareMathOperator{\esssup}{ess\,sup}
\DeclareMathOperator{\sen}{sen}
```

Ya en el documento podemos invocar el operador:

```
\rank{x} rank x
\esssup{x} ess sup x
\sen{x} sen x
```

3.1.5. Paréntesis

Para ingresar paréntesis en el modo matemático tenemos los comandos `\left` y `\right` seguido del tipo de paréntesis que queremos, por ejemplo `\left(` o `\left[`. De esta forma se ajusta el tamaño del paréntesis al tamaño del texto que estamos ingresando.

Delimitador	normal	<code>\left</code> <code>\right</code>	<code>\bigl</code> <code>\bigr</code>	<code>\Bigl</code> <code>\Bigr</code>	<code>\biggl</code> <code>\biggr</code>	<code>\Biggl</code> <code>\Biggr</code>
	$a\left(\frac{a}{b}\right)$	$a\left(\frac{a}{b}\right)$	$a\left(\frac{a}{b}\right)$	$a\left(\frac{a}{b}\right)$	$a\left(\frac{a}{b}\right)$	$a\left(\frac{a}{b}\right)$

Si queremos poner llaves `{}` podemos usar `\left\{` y `\right\}`, para que se adapte al tamaño del texto, o elegir el tamaño con `\big\{`. Si solamente queremos agregar uno, por ejemplo el izquierdo, tenemos que escribir `\left\{` y luego cerrarlo con `\right.`; del mismo modo, hay que poner: `\bigl\{` y `\bigr.`

3.1.6. Fracciones y fracciones anidadas

Para agregar fracciones en el modo matemático tenemos el comando:

`\frac{numerador}{denominador}`.

Si lo que queremos es concatenar fracciones, y que el tamaño de lo que agregamos no se vaya achicando, usamos por ejemplo

```
\cfrac{1}{\sqrt{2}+}
\cfrac{1}{\sqrt{2}+}
\cfrac{1}{\sqrt{2}+\dotsb}
}}
```

$$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}}$$

3.1.7. Alineación de símbolos

Supongamos que queremos poner dos símbolos pero uno sobre otro, por ejemplo \xrightarrow{n} . Para eso tenemos el comando `\stackrel{arriba}{abajo}`, observemos que lo que va arriba aparece mas pequeño que lo de abajo. Si, por ejemplo, queremos poner dos índices en una sumatoria, este comando no sería apropiado, y tendríamos que hacer (suponiendo incluido en el preámbulo el paquete `amsmath`):

```
\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i, j)
```

$$\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i, j)$$

3.2. Entornos matemáticos

Alguno de los entornos matemáticos más comunes, son:

```
\begin{multline}
a+b+c+d+e+f \\
+i+j+k+l+m+n
\end{multline}
```

$$a + b + c + d + e + f + i + j + k + l + m + n \quad (3.1)$$

```
\begin{gather}
a_1=b_1+c_1 \\
a_2=b_2+c_2-d_2+e_2
\end{gather}
```

$$a_1 = b_1 + c_1 \quad (3.2)$$

$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (3.3)$$

```
\begin{align}
a_1&=b_1+c_1 \\
a_2&=b_2+c_2-d_2+e_2
\end{align}
```

$$a_1 = b_1 + c_1 \quad (3.4)$$

$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (3.5)$$

```
\begin{align}
a_{11}&=b_{11}&
a_{12}&=b_{12} \\
a_{21}&=b_{21}&
a_{22}&=b_{22}+c_{22}
\end{align}
```

$$\begin{array}{lcl}
 a_{11} = b_{11} & a_{12} = b_{12} & \\
 a_{21} = b_{21} & a_{22} = b_{22} + c_{22} & (3.6)
 \end{array}$$

```
\begin{flalign*}
a_{11}&=b_{11}&
a_{12}&=b_{12} \\
a_{21}&=b_{21}&
a_{22}&=b_{22}+c_{22}
\end{flalign*}
```

$$\begin{array}{lcl}
 a_{11} = b_{11} & a_{12} = b_{12} & \\
 a_{21} = b_{21} & a_{22} = b_{22} + c_{22} &
 \end{array}$$

3.3. Matrices

3.3.1. Entorno array

Para generar matrices tenemos varios entornos. Vamos a ver primero el entorno `\begin{array}` que es similar al entorno `tabular` ya visto antes.

```
\begin{equation*}
\mathbf{X} = \left(
\begin{array}{ccc}
x_1 & x_2 & \dots \\
x_3 & x_4 & \dots \\
\vdots & \vdots & \ddots
\end{array}
\right)
\end{equation*}
```

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots \\ x_3 & x_4 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Como en el entorno `tabular`, al comienzo tenemos que indicar cuántas columnas vamos a agregar, y en qué posición irá el texto dentro de esa columna. Luego agregamos las columnas separándolas por `&`. Al no ser un entorno matemático, este entorno debe ir incluido dentro de un entorno matemático. En el ejemplo anterior `\begin{equation*}`. Podemos usar el entorno `array` para crear funciones “partidas”, por ejemplo:

```
\begin{equation*}
|x| = \left\{
\begin{array}{rl}
-x & \text{si } x < 0, \\
0 & \text{si } x = 0, \\
x & \text{si } x > 0.
\end{array}
\right.
\end{equation*}
```

$$|x| = \begin{cases} -x & \text{si } x < 0, \\ 0 & \text{si } x = 0, \\ x & \text{si } x > 0. \end{cases}$$

Para este último caso se puede usar también el entorno `cases`, siempre que se haya incluido en el preámbulo el paquete `amsmath`.

3.3.2. Otros entornos

El conjunto de paquetes \mathcal{AMS} trae entornos que permiten crear matrices de forma más fácil, con diferentes delimitadores. Por ejemplo `matrix` sin delimitador, `pmatrix` con `(`, `bmatrix` con `[`, `Bmatrix` con `{`, `vmatrix` con `|`, o `Vmatrix` con `||`. Por ejemplo

```

\begin{equation*}
\begin{matrix}
1 & 2 \\
3 & 4
\end{matrix} \quad \qquad \qquad
\begin{Vmatrix}
p_{11} & p_{12} & p_{1n} \\
p_{21} & p_{22} & p_{2n} \\
\vdots & \vdots & \vdots \\
p_{m1} & p_{m2} & p_{mn}
\end{Vmatrix}
\end{equation*}

```

En estos entornos, al igual que en el entorno `array`, se pueden usar los comandos `\multicolumn` y `\multirow` que se usaron en el entorno `tabular`.

3.4. Teoremas, Lemas, etc

3.4.1. Teoremas y demostraciones

En esta sección vamos a ver cómo definir entornos para enunciar Teoremas, y que estos sigan cierta numeración. Empecemos con un ejemplo, supongamos que queremos definir un entorno `\begin{teo}` que al invocarlo escriba la palabra **Teorema** y que siga la numeración del capítulo en el que se encuentra. Para eso escribimos en el preámbulo:

```
\newtheorem{teo}{Teorema}[chapter]
```

Si en lugar de que siga la numeración del capítulo queremos que siga la de la sección cambiamos `chapter` por `section`. La demostración se escribe entre `\begin{proof}` y `\end{proof}`, lo cual hace que la demostración quede encerrada entre la palabra “demostración”, o “proof”, y un símbolo distintivo.

```

\begin{teo} Sea  $x \in \mathbb{R}^2 \dots$ 
\begin{proof}
\end{proof}
\end{teo}

```

Teorema 3.1. *Sea $x \in \mathbb{R}^2 \dots$*

Demostración. □

El símbolo al final de la demostración se puede cambiar con el comando:

```
\renewcommand{\qedsymbol}{símbolo}.
```

3.4.2. Lemas

Si queremos definir un entorno que permita escribir lemas, y que por defecto siga la numeración de los teoremas, escribimos `\newtheorem{lem}[teo]{Lema}` en el preámbulo. Por ejemplo:

```
\begin{lem} Sea  $f$  continua
\end{lem}
```

Lema 3.2. *Sea f continua*

Observemos que el lema se numera con un **3.2** donde el 3 indica el capítulo y el 2 se debe a que continúa la numeración del teorema. Si ahora hacemos:

```
\begin{lem} Sea  $f$  continua
\end{lem}
```

Lema 3.3. *Sea f continua*

3.4.3. Corolarios, Proposiciones, etc

De forma análoga se pueden definir los entornos para los corolarios y proposiciones, por ejemplo:

```
\newtheorem{prop}[teo]{Proposición}
\newtheorem{obs}[teo]{Observación}
\newtheorem{cor}[teo]{Corolario}
```

Estos entornos siguen la numeración de los teoremas. Supongamos que en el preámbulo hacemos `\newtheorem{obs}[subsection]{Observación}`

```
\begin{obs} Sea  $f$  continua
\end{obs}
\begin{obs} Otra observación
\end{obs}
```

Observación 3.4.4. *Sea f continua*

Observación 3.4.5. *Otra observación*

Ahora las observaciones siguen la numeración de la subsección donde se encuentran. En este caso, como estamos en la subsección 3.4.3, continúan a 3.4.4 y 3.4.5.

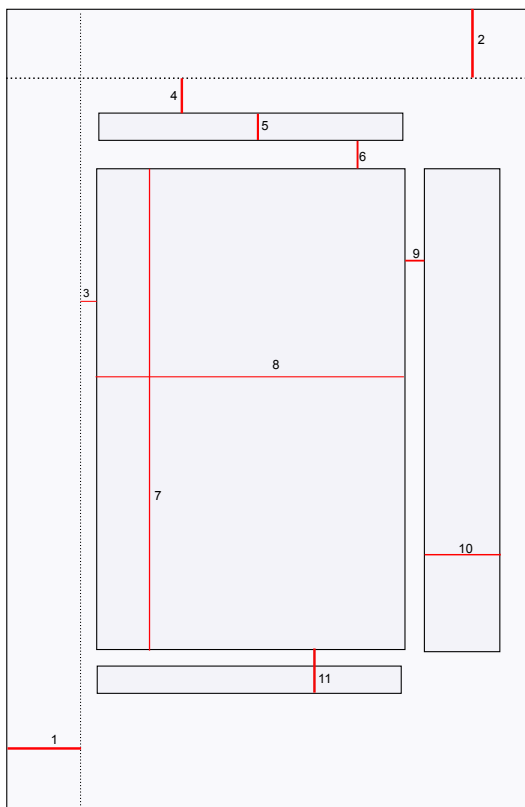
Si usamos el paquete `amsthm`, declarado en el preámbulo, podemos cambiar el formato en el que aparecerá el texto dentro de un Teorema, Lema, etc. Para eso hay que agregar, antes de definirlo, el comando `\theoremstyle{definition}` si queremos que el título del entorno aparezca en negrita y el resto sin formato, `\theoremstyle{plain}` si queremos el título en negrita, y el resto en cursiva, o `\theoremstyle{remark}` para poner el título en cursiva, y el resto sin formato.

Capítulo 4

Encabezados y márgenes

En este capítulo vamos a configurar márgenes, por medio del paquete *fancyhdr*. Además del paquete tenemos que agregar en el preámbulo el comando `\pagestyle{fancy}`.

4.1. Márgenes



- 1: Una pulgada + `\hoffset`
- 2: Una pulgada + `\voffset`
- 3: `\oddsidemargin`
- 4: `\topmargin`
- 5: `\headheight`
- 6: `\headsep`
- 7: `\textheight`
- 8: `\textwidth`
- 9: `\marginparsep`
- 10: `\marginparwidth`
- 11: `\footskip`

Los valores anteriores pueden ser negativos, y las unidades: in = Pulgadas, cm = Centímetros, m = Milímetros, pt = Puntos. Por ejemplo, el documento actual tiene como márgenes:

```
\oddsidemargin 0.3cm
\textwidth= 17cm
\textheight= 25.5cm
\headsep= 0.5cm
\hoffset= -1cm
\voffset= -2cm
```

4.2. Encabezado y pie de página

4.2.1. Encabezado

Para configurar el encabezado tenemos tres comandos que deben ir en el preámbulo:

`\lhead{}` Para indicar qué irá a la izquierda.
`\chead{}` Para indicar qué irá al centro.
`\rhead{}` Para indicar qué irá a la derecha.

Supongamos que a izquierda queremos poner el número de página, en el centro el nombre del capítulo en minúscula y negrita y a derecha el de la sección. Eso lo hacemos con:

```
\lhead{\thepage}, \chead{\bfseries \nouppercase \leftmark}, \rhead{\rightmark}
```

Si queremos que aparezca una línea horizontal y elegir el tamaño, supongamos 2pt, tenemos el comando:

```
\renewcommand{\headrulewidth}{2pt}
```

4.2.2. Pie de Página

Para configurar el pie de página tenemos tres comandos análogos:

`\lfoot{}` Para indicar qué irá a la izquierda.
`\cfoot{}` Para indicar qué irá al centro.
`\rfoot{}` Para indicar qué irá a la derecha.

Supongamos que al pie queremos colocar: en el centro el número de página y el total, y a la derecha un texto. Para eso:

```
\cfoot{ \thepage \ de \ \pageref{LastPage}}
\rfoot{texto}
```

Análogamente, si queremos agregar una línea divisoria de 2pt:

```
\renewcommand{\footrulewidth}{2pt}
```

4.2.3. Encabezados distintos para páginas pares e impares

Vamos a usar la siguiente notación:

E	Par
O	Impar
L	Izquierda
C	Centro
R	Derecha
H	Encabezado
F	Pie de página

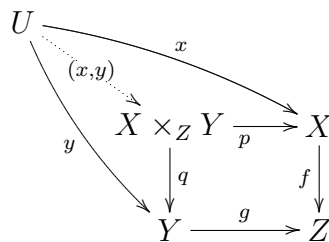
Y por lo tanto para cambiar los encabezados y pie de página podríamos hacer por ejemplo:

```
\fancyhead[RO,LE]{\bfseries \leftmark}
\fancyfoot[LE,RO]{\thepage}
\fancyfoot[LO,CE]{Gu\`ia de Latex}
\fancyfoot[CO,RE]{\slshape \rightmark}
```

Capítulo 5

Diagramas

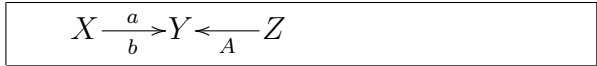
En este capítulo vamos a ver cómo crear diagramas con el paquete `Xy`, por ejemplo de la forma:



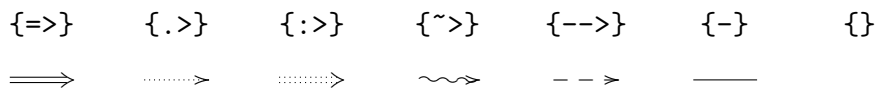
5.1. Diagramas de una fila

En general los diagramas van a empezar con el comando `\xymatrix@1{}`, si éste va a tener una sola fila, o `xymatrix{}` si va a tener más. Empecemos con un ejemplo sencillo:

```
\xymatrix@1{X\ar[r]^a_b  
& Y & Z\ar[l]_A }
```

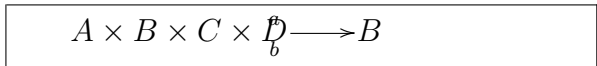


Del ejemplo anterior vemos que `\ar` indica que vamos a poner una flecha, y `[r]` hacia la derecha, luego con `^` indicamos qué irá sobre la flecha - podría no haber nada más que la flecha - y con `_` qué irá debajo. Observemos que el origen y el destino de la flecha se separan con `&`, y que la flecha que parte de `Z`, si bien es `^A`, en este caso la `A` aparece abajo. Podemos cambiar el estilo de la flecha que usamos con `\ar@{estilo}[direccion]`, donde en `dirección` podemos poner `r` (right), `l` (left) - para `\xymatrix{}` tenemos además: `u` (up) y `d` (down)- y en `estilo` podemos poner alguno de los siguientes:



Los símbolos que se colocan sobre y debajo de las flechas, se alinean de modo de quedar en el punto medio entre los símbolos precedente y siguiente a la flecha, lo cual produce:

```
\xymatrix@1{A\times B\times C\times D  
\ar[r]^a_b & B }
```



Para corregir ese problema, tenemos que indicarle en qué posición queremos que vayan los símbolos. Para eso podemos elegir nosotros la posición:

```
\xymatrix@1{A\ar[r]^{\langle + \rangle}&B}
\xymatrix@1{A\ar[r]^{\langle + \rangle}&B}
\xymatrix@1{A\ar[r]^{\langle - \rangle}&B}
\xymatrix@1{A\ar[r]^{\langle .6 \rangle}&B}
```



Si lo que queremos es poner un *corte* en la flecha e ingresar un símbolo ahí, podemos hacer:

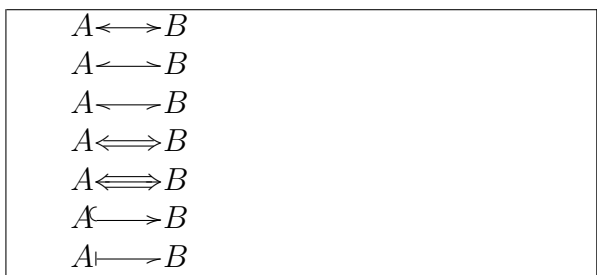
```
\xymatrix@1{A\ar@{.}>[r]|f&B}
\xymatrix@1{A\ar@{.}>[r]|>\{f\}&B}
\xymatrix@1{A\ar@{.}>[r]|<\{f\}&B}
\xymatrix@1{A\ar[r]|\hole & B }
```



5.1.1. Variantes de flechas

Otras posibles variantes (que también sirven para el caso en que hay más de una fila) son:

```
\xymatrix@1{A\ar@{<->}[r]&B}
\xymatrix@1{A\ar@{<->}^{}[r]&B}
\xymatrix@1{A\ar@{<->}_{}[r]&B}
\xymatrix@1{A\ar@{<->}@2{}[r]&B}
\xymatrix@1{A\ar@{<->}@3{}[r]&B}
\xymatrix@1{A\ar@{<->}^{\{ \} }[r]&B}
\xymatrix@1{A\ar@{<->}|_{- \{ \} }[r]&B}
```

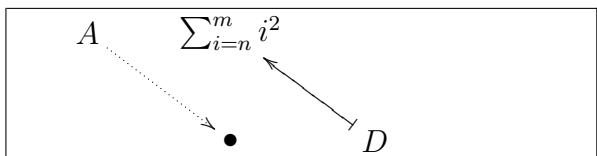


Observemos que en los últimos dos ejemplos hemos construido la forma que tendrá la flecha indicando primero qué irá en el extremo izquierdo. Por ejemplo $\hat{\{ \}}$ para decir que queremos que vaya un arco, pero elevado sobre el nivel del resto de la flecha, luego $-$ para indicar que en el medio será una línea, y finalmente $>$ para indicar el símbolo que irá al final.

5.2. Diagramas de 2 o más filas

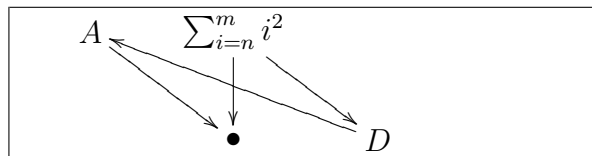
Si queremos hacer un diagrama de más de una fila, por ejemplo el que aparece al comienzo de este capítulo, lo que hacemos es usar el comando `\xymatrix`. Vamos a empezar con un ejemplo simple:

```
\xymatrix{
A\ar@{.}>[dr] & \{\sum_{i=n}^m i^2\} \\
& \bullet & \leftarrow D}
& \{\bullet\} & D \ar@{|->}[ul]}
```



Observemos que la sintaxis es similar a la de un *array*, donde cada columna se separa por `&`, y cada fila por `\\`. Las flechas ahora las colocamos con el comando `\ar @{estilo}[posición]`, donde en *estilo* se pueden elegir los estilos que se vieron para el caso de una fila, y en *posición* indicamos cuál será el final. En el ejemplo la flecha que parte de *D* sube una fila, por eso el *u*, y se desplaza hacia la izquierda un lugar, de ahí la *l*. Si quisiéramos que se desplazara dos lugares hacia la izquierda y uno hacia arriba pondríamos: `\ar [ul1]`. Por ejemplo:

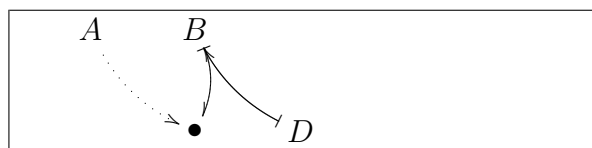
```
\xymatrix{
A \ar[dr] & \sum_{i=n}^m i^2 \\
\ar[d] \ar[dr] & \bullet \\
& D \ar[ull]}
```



5.2.1. Otras flechas

Supongamos que queremos poner flechas curvadas, como en el primer ejemplo de este capítulo. Para eso tenemos que agregar `@/_/` para curvar hacia abajo, `@/^/` para curvar hacia arriba, o si queremos curvarla aún más en alguna dirección podemos hacer `@/_1pc/`. Estos parámetros pueden combinarse con los anteriores, por ejemplo:

```
\xymatrix{
A \ar@{.}>@/_/[dr] & \\
B \ar@{|-_{>}}/^/[d] & \bullet \\
& D \ar@{|->}[ul]}
```



Ahora ya es claro que el código para el ejemplo del comienzo del capítulo es:

```
\xymatrix{
U \ar@/_/[ddr]_y \ar@/^/[drr]^x \\
\ar@{.}>[dr]|-(x,y) \\
& X \times_Z Y \ar[d]^q \ar[r]_p \\
& X \ar[d]_f \\
& Y \ar[r]^g & Z}
```


Capítulo 6

Otros tópicos

6.1. Objetos Flotantes

Los objetos flotantes son objetos como por ejemplo tablas, o imágenes, que se caracterizan por el hecho de que no pueden ser *partidos* cuando se termina una página. Dichos objetos tienen que ser ubicados en una posición dada. Vamos a referirnos en esta sección solamente al entorno `\begin{table}[pos] \end{table}`, que típicamente se usa para ingresar tablas, y luego poder hacer con ellas un índice de tablas, y al entorno `\begin{figure}[pos]\end{figure}` que permite agregar figuras. En `pos` ponemos la posición de la página donde va a ir la figura. Tenemos entre ellas:

h (here)	En el lugar donde se agrega la imagen o tabla.
t (top)	En la parte superior de la página.
b (bottom)	En la parte inferior de la página.
p	En una página determinada.

Podemos poner más de una de las opciones anteriores, lo cual significa que \LaTeX puede optar entre ellas.

Veamos un ejemplo para el entorno `table`:

```
\begin{table}[htb]
\begin{tabular}{c}
Tabla
\end{tabular}
\caption{nombre de la tabla}
\end{table}
```

El comando `\caption{}` le agrega un nombre a la tabla; dicho nombre aparecerá en el listado de tablas.

6.2. Insertar imágenes

Supongamos que queremos insertar la imagen `figura.pdf`, centrada, reescalada 50 %, y que lleve un título dado; en el preámbulo incluimos el paquete `graphicx`, y entonces usamos:

```
\begin{figure}[ht]
\begin{center}
\includegraphics[scale=.5]{figura.pdf}
\caption{Nombre de la imagen}
```

```
\end{center}
\end{figure}
```

6.3. Índices

L^AT_EX genera de forma automática el índice, simplemente agregando en el lugar que se desee que éste aparezca el comando:

```
\tableofcontents
```

Si queremos que un determinado capítulo o sección no aparezcan simplemente agregamos dicha sección con un asterisco, por ejemplo `\section*{}`. Los siguientes comandos crean respectivamente un listado de tablas y un listado de figuras: `\listoftables` y `\listoffigures`.

Si queremos agregar una entrada a la lista de capítulos, tablas, o figuras, con un determinado título, agregamos además el comando:

```
\addcontentsline{lugar}{unidad}{nombre}.
```

En `lugar` indicamos en qué lugar vamos a agregar la entrada; será `toc` si queremos agregarla como una entrada en la lista de capítulos, `lof` para agregarla a la lista de figuras, o `lot` para agregarlo a la lista de tablas. Este comando se puede utilizar en general para agregar cosas a esas listas. En `unidad` indicamos si lo vamos a agregar como un capítulo aparte (`chapter`), como una sección (`section`) etc, y `nombre` es el nombre con el que se agregará.

6.4. Bibliografía y apéndices

6.4.1. Bibliografía

La bibliografía se agrega entre los comandos:

```
\begin{thebibliography}{99}
\end{thebibliography}
```

Cada ítem de la bibliografía se agrega con el comando `\bibitem {etiqueta}`, donde `etiqueta` será el nombre con el cual identificaremos esa referencia en nuestro documento, por medio del comando `\cite{etiqueta}`.

6.4.2. Apéndice

Para agregar apéndices (esto es, capítulos que aparecerán en el índice con letras en lugar de números, y que comenzarán con la palabra apéndice), simplemente agregamos el comando `\appendix`, antes de los capítulos que serán tomados como apéndices, por ejemplo:

```
\appendix
\chapter{Repaso de...}
```

6.5. Crear comandos

Para crear un comando, en el preámbulo del documento usamos la sentencia:

```
\newcommand{\nombre}[num]{definicion}
```

Aquí el `nombre` es el nombre del comando que estamos creando, nos referiremos a él como `\nombre`; `num` es el número de parámetros que tendrá nuestro comando (de 0 a 9); y en `definición` irá la definición del comando que estamos creando.

Una de las aplicaciones más simples de los comandos nuevos creados por nosotros es abreviar un comando ya existente. Supongamos que queremos abreviar el comando `\longleftarrow` (\longleftarrow); para eso agregamos

```
\newcommand{\lra}{\longleftarrow}
```

Si en el documento ahora hacemos `$$\lra`, se produce: \longleftarrow . Supongamos que queremos crear un comando que nos permita escribir funciones de modo de pasarle como parámetros el dominio y el codominio, es decir, por ejemplo: $f : A \longrightarrow B$. Para eso creamos:

```
\newcommand{\funcion}[2]{#1 \longrightarrow #2}
```

y nos referimos a ella como `$$\funcion{A}{B}`, por ejemplo $f : A \longrightarrow B$.

Bibliografía

- [1] M. Goosens, F. Mittlebach, A. Samarin, *The L^AT_EX Companion*, Addison-Wesley, 1994.
- [2] Helmut Kopka, Patrick W. Daly, *A guide to L^AT_EX*, Addison-Wesley, 1999.
- [3] T. Oetiker, *The not so Short Introduction to L^AT_EX*, <http://www.ctan.org/tex-archive/info/lshort/english/>.