

# Tutorial Share L<sup>A</sup>T<sub>E</sub>X

Gabriel De Simone

Argentina, Julio 2015

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Expresiones matemática</b>	<b>3</b>
2.1. Símbolos matemáticos (y tablas)	3
2.2. Subíndices y superíndices	3
2.3. Corchetes y paréntesis	4
2.4. Teoremas y pruebas	4
2.5. Definiciones	5
<b>3. Operadores</b>	<b>5</b>
3.1. Límites	5
3.2. Operadores de referencia	5
<b>4. Imágenes</b>	<b>5</b>
4.1. Insertar imágenes, cambiarles el tamaño y rotarlas	5
4.2. Posicionar imagen	6
4.3. Referencias y epígrafes	7
<b>5. Formato</b>	<b>8</b>
5.1. Negrita, subrayado, cursiva y énfasis	8
5.2. Encabezados y pies de páginas	8
5.3. Formato del párrafo	8
5.3.1. Sangría	8
5.3.2. Espacio entre párrafos	8
5.3.3. Interlineado	9
5.3.4. Justificado, alineado a izquierda, alineado a derecha y centrado	9
5.4. Uso de colores	9
5.5. Referenciar secciones y ecuaciones	9
5.6. Tamaños y tipos de letra	10
<b>6. Referencias Bibliográficas</b>	<b>10</b>
<b>Las Referencias fantasía</b>	<b>10</b>
Artículos fantasía	10
<b>7. Paquetes específicos útiles</b>	<b>11</b>
7.1. Paquete circuitos	11
7.2. Diagramas simples	13
<b>8. Bibliografía</b>	<b>16</b>
Internet	16

## Resumen

En el presente escrito se da una breve introducción a la manera en la que latex funciona partiendo del supuesto de que el lector es un ignorante en lo respecta a esta forma de trabajar. Se muestra y explica la utilización de diversas herramientas que suelen ser de utilidad a la hora de escribir un informe. El actual desarrollo se enfoca particularmente en la escritura de artículos (entiéndase informes o escritos académicos para leer en “una sentada”) aunque la inmensa mayoría de los lineamientos expuestos son válidos para otros tipos de estructuras como libros o cartas. La motivación del escrito es la no la existencia de la información aquí tratada de forma compacta y en español; la que abunda se encuentra dispersa. El lector podrá apreciar la manera de generar un artículo, al observar simultáneamente el código asociado y el resultado obtenido. No se han utilizado plantillas.

## 1. Introducción

L<sup>A</sup>T<sub>E</sub>X es un procesador de texto de software, muy utilizado para la composición de literaturas académicas dado que la calidad tipográfica de los documentos obtenidos rivaliza con la de las editoriales científicas. Permite la incorporación de secciones, índices y fórmulas de manera particularmente sencilla y ordenada, facilitando que usted se concentre en el contenido y no en el formato. Este último puede modificarse de manera general con algunas líneas de código sin tener que ir por párrafo como en otros procesadores y evita el tener que realizar constantemente ajustes de este tipo.

Nos centraremos particularmente en Share L<sup>A</sup>T<sub>E</sub>X, que básicamente es lo mismo pero se diferencia por permitir el trabajo online sobre un documento en forma colaborativa y en simultáneo; esto no quita que todo lo tratado sea válido para otros soportes TEX.

Antes de comenzar con la guía en sí, es de menester hacer algunas aclaraciones. Primero, el desarrollo posterior se ha basado en [5] y una información mucho más amplia que la que aquí se expone puede ser consultada en ese sitio web. Segundo, se nombra a lo largo del texto otras bibliografías de fantasía; no se han usado en esta literatura y son solo ilustrativas. Tercero, todo lo que se muestra tiene una manera alternativa de hacerse. Cuarto, el documento busca introducir al lector de manera práctica y no se centra en cuestiones teóricas. Quinto, existen ciertos símbolos que son reservados por latex para introducir comandos, a los mismos debe anteponerles una barra derecha para imprimirlos en el PDF; algunos de estos son: #, \$, %, &, - y {.

Para comenzar se debe tener en cuenta que el trabajo en L<sup>A</sup>T<sub>E</sub>X se divide en tres etapas. La primera consiste en escribir el código (archivo de extensión TEX; incluye al trabajo propiamente dicho y las líneas asociadas al formato, imágenes, etc); la segunda en compilarlo, que implica pasar a PDF lo escrito y la tercera abrir el PDF. Share L<sup>A</sup>T<sub>E</sub>X permite realizar las tres operaciones en simultáneo. Una vez en la página, cuando se crea un nuevo proyecto, se tiene la pantalla dividida; en la parte izquierda va el código y en la derecha se va mostrando el PDF a medida que recompilamos. Le recomiendo mirar en simultáneo el PDF y el TEX del presente tutorial (que se encuentra al final del documento) para comprenderlo completamente, pues no se explica en el primero lo que aparece como comentario en el código, además de otros detalles.

La parte que requiere estudio y que se trata seguidamente es la relativa al código. Todo documento trabajado en latex consta de dos secciones bien visibles en el archivo TEX. La primera es el llamado preámbulo en el que se define el tipo de texto con el que se va a trabajar, se llama a ciertos paquetes que incorporan funciones que latex no trae por defecto, se definen algunos aspectos del formato y se colocan algunos datos como el autor, la fecha; llega en el código hasta la línea “begin{document}”. Al igual que al programar en muchos otros lenguajes, debe indicarse el fin y el inicio de ciertas estructuras; en el caso del cuerpo del documento, que es la otra parte del código, su fin debe indicarse al terminar el trabajo con “end{document}”. Estas estructuras que comienzan y terminan de manera explícita, indicando el inicio de algún comando o función en particular, se denominan ambientes y vienen a ser semejante a las sentencias if, for o while de otros lenguajes en este sentido.

En el preámbulo se han colocado toda una serie de paquetes que se irán utilizando a lo largo del texto y se explicarán oportunamente. Ahora resulta oportuno tratar la primera línea de todo trabajo en latex: documentclass{clase de documento}. Esta línea define el formato básico que se utilizará. Puede colocarse artículo (article), libro (book), informe (report), entre tantos otros de acuerdo a las necesidades. Determina cosas como el tamaño de las sangrías, espacios entre títulos o la forma de numerar elementos según el tipo de texto, basándose en las estructuras de más uso en esos escritos. Muchos de estos valores por defecto pueden modificarse (lo haremos) e incluso usted puede crear su propio “documentclass como se enseña en diversas bibliografías. Es necesario colocar esta línea porque le da orden a todo lo que usted no ajuste manualmente y permite tener una base desde dónde comenzar a trabajar.

Una vez que se da inicio al cuerpo del texto, puede fácilmente incorporar secciones, subsecciones y capítulos, que se numeran automáticamente y aparecen con el mismo formato según el caso. Su estudio se realiza al progresar en el documento. Se aclara desde ahora que Latex cuenta con distintos niveles y subniveles para organizar contenidos que son desde más a menos general: parte (part), capítulo (chapter), sección (section), subsección (subsection), subsubsección (subsubsection), párrafo (paragraph) y subpárrafo (subparagraph); todos se inician anteponiendo una barra invertida. No todos están disponibles en el formato artículo.

## 2. Expresiones matemática

Por ser de las principales ventajas y como a los ingenieros técnicamente nos importa la matemática, es importante poder expresarla y escribir fórmulas en latex. Veremos cómo hacerlo.

Para colocar una fórmula en línea con el texto puede usarse la siguiente estructura  $x^2 + y^2 = z^2$ . En cambio para hacerlo al estilo word, podemos hacer así:

$$x^n + y^n = z^n$$

Es el famoso teorema de Pitágoras generalizado, que como sabrá no es válido.

La ecuación debe ser incluida dentro del ambiente `equation` si se requiere que aparezca numerada. Con la siguiente estructura queda prolija y se muestra cómo colocar una fracción.

$$A = \frac{\pi r^2}{2} = \frac{1}{2} \pi r^2 \quad (1)$$

### 2.1. Símbolos matemáticos (y tablas)

Existen diversos y la forma de llamarlos en latex se coloca en la siguiente tabla, y de paso le enseño a poner una tabla.

Descripción	código
Letras griegas	$\alpha\beta\gamma\rho\sigma\delta\epsilon$
Operadores binarios	$\times \otimes \oplus \cup \cap$
Operadores de relación	$\subset \supset \subseteq \supseteq$
Otros importantes	$\int \oint \Sigma \Pi$

Tabla 1: Tabla de símbolos

Cuando en una columna de una tabla se debe insertar un texto particularmente extenso, el ambiente “`tabular`” no puede resolver la situación de manera conveniente y la tabla escapa al ancho de la página. Un ambiente que soluciona el caso es el llamado “`tabularx`”, que se utiliza de la misma manera, salvo que en las columnas afectadas por estas longitudes de texto infrecuentemente grandes, debe colocarse una “X”, al declarar la estructura básica de la tabla. Un ejemplo se muestra a continuación. Debe incorporarse en el preámbulo el paquete “`usepackage tabularx`” pues este ambiente no lo trae latex por defecto.

Debe tenerse en cuenta que este ambiente no es compatible con el comando “`caption`”, por lo que aquí se muestra una forma de colocar un epígrafe a estas tablas. La trampa consiste en colocar un ambiente “`tabular`” después de la tabla, sin una tabla dentro; se muestra seguidamente.


Nombre	Símbolo	Activo/pasivo	Características	Aplicaciones
Resistor		Pasivo	Se opone al paso de corriente, generando una caída de tensión que se traduce en potencia disipada. Existen de materiales diversos que constan de resistividad variable con la temperatura y con la que la resistencia varía proporcionalmente, dependiendo también de la geometría. Cumplen la ley de Ohm. Su efecto se cuantifica en Ohmios.	Control de volumen y tono en los dispositivos de sonido, control de brillo, etc. Existe un tipo de potenciómetro que se fabrica especialmente para ajustar voltajes o corrientes en algunos circuitos integrados.

Tabla 2: Componentes de circuito

### 2.2. Subíndices y superíndices

Usted podría pensar que son algo secundarios, estando equivocado, pues realmente los usamos a menudo y hacerlos con word es realmente engorroso. Los mismos se escriben tan solo anteponiendo un símbolo al índice tal como veremos enseguida.

Por ejemplo, para una integral definida podemos hacer lo siguiente

$$\int_0^1 x^2 dx$$

Se ve claramente que el “guión” coloca el subíndice y el “piquito (diría una vieja)” el superíndice. Si los mismos son largos puede hacerse de la siguiente manera que es muy cómoda.

$$x^{2\alpha} - 1 = y_{ij} + y_{ij}$$

Este último que le pongo ahora es realmente excelente porque las sumatorias se usan bastante y las fracciones más todavía, también aparece el símbolo del infinito.

$$\sum_{i=1}^{\infty} \frac{1}{n^s} = \prod_p \frac{1}{1 - p^{-s}}$$

### 2.3. Corchetes y paréntesis

Es la forma de colocar las matrices y los vectores, que tan poco usamos a decir verdad, pero vale la pena ponerlo. Los mismos pueden ajustarse de manera automática o manual. Primero la automática, que se basa en construir una tabla como la de antes y colocarle el símbolo en cuestión a la derecha y la izquierda.

$$\left\{ \begin{array}{ccc} 1 & 5 & 8 \\ 0 & 2 & 4 \\ 3 & 3 & -8 \end{array} \right\}$$

En forma manual se deben colocar los siguientes símbolos que se presentan en la tabla [5]

Descripción	código
Paréntesis	$((((($
Corchetes	$]]]]]$
Llaves	$\{\{\{\{\}$

Y para cerrar un ejemplo clásico  $(90 + 76z)$

### 2.4. Teoremas y pruebas

Para dar un formato particular a estos, debemos primero definir cómo los llamaremos y la palabra que aparecerá en el PDF cuando se lo haga, lo mismo se hace en el preámbulo a través del comando `\newtheorem{theorem}{Teorema}`, que tiene dos parámetros. El primero es el nombre del ambiente definido (algunos ambientes es necesario definirlos) y el segundo, el nombre con el que aparece en el texto. El `[theorem]` que aparece en las dos líneas siguientes a esa, indica que el contador de los corolarios y lemas debe reiniciarse al iniciar un nuevo teorema; si en cambio, se quiere que usen el mismo que los teoremas, debe colocarse antes del último de los dos textos entre llaves en esas líneas y si se quiere que tengan su propia numeración no debe colocarse.

**Teorema 1.** *Dios todo lo puede.*

**Teorema 2** (Teorema generalizado). *Es la generalización del teorema anterior. Dios todo lo puede, cuando quiere y como quiere.*

Las consecuencias del teorema 2 se ven en la realidad que lo rodea.

**Corolario 2.1.** *No desafío a Dios.*

**Lema 2.1.** *Dadas dos personas, harán ellas lo que Dios desee, no su libertad.*

*Demostración.* No necesito más prueba que el antiguo testamento. □

## 2.5. Definiciones

Una manera prolija de dar definiciones es usando el formato de los teoremas que realmente queda maravilloso. El paquete `amsthm` permite introducir esta función, **y también el formato anterior para demostraciones**. Los otros comandos `theoremstyle{definition}` y `newtheorem{definition}{Definición}[section]`, permiten crear un ambiente (`definition`) con el estilo de los teoremas, darle un nombre con el que aparecerá en el texto y reiniciar su contador en cada sección.

**Definición 2.1.** Chiripioca Dícese del estado de una persona iracunda y desenchajada. Ej: Le agarró la...

## 3. Operadores

Los operadores tienen una notación particular en latex. Se destacan algunos.

### 3.1. Límites

Para colocarlos tan prolijo como en el stuart o en el eduart tan solo se hace lo siguiente, se considera que el código se explica a sí mismo.

$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} = \frac{df(x)}{dx}$$

Que queda de esta forma si se usa en línea con el texto  $\lim_{x \rightarrow h}(x - h)$ .

### 3.2. Operadores de referencia

Para tenerlos a mano, se adjuntan en la tabla siguiente.

Descripción	código
Seno	<code>sin</code>
Arcoseno	<code>arcsin</code>
Seno hiperbólico	<code>sinh</code>

Son bastante intuitivos y los demás se toman de modos semejantes.

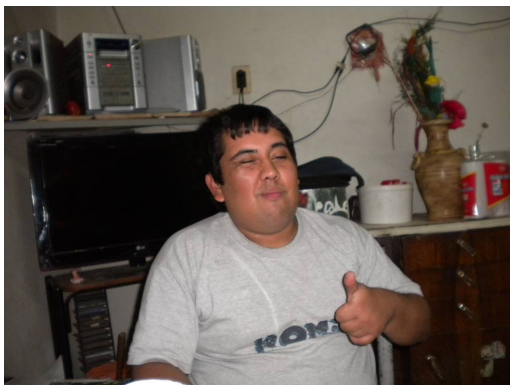
## 4. Imágenes

Un buen trabajo pasa a ser excelente cuando se agrega la cantidad de imágenes correcta de adecuada forma, prolijamente, con el tamaño adecuado y sus epígrafes. Esto nos ocupa ahora mismo.

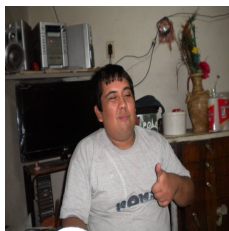
### 4.1. Insertar imágenes, cambiarles el tamaño y rotarlas

Latex no tiene la capacidad de manipular imágenes por sí mismo, por esa razón se debe importar el paquete `graphicx` en el preámbulo. Para usar dicho paquete se añadió al inicio la siguiente línea `“\usepackage{graphicx}”`. El comando `“\graphicspath{ {images/} }”` le indica a latex que las imágenes están guardadas en una carpeta llamada `images` bajo el directorio actual, las mismas deben subirse a Sharelatex para trabajar, esto se hace con el ícono de la parte superior izquierda de la pantalla.

El comando `“\includegraphics{gordo}”` es el que se encarga de incluir la imagen en el documento. El nombre del archivo de imagen no debe contener espacios ni caracteres especiales.



El tamaño puede modificarse con el comando `scale`. También es posible definir de forma arbitraria el ancho y el alto.



Como probablemente ya ha adivinado, los parámetros dentro de los corchetes [`width=3cm`, `height=4cm`] definen el ancho y alto de la imagen. Pueden usarse diferentes unidades de medida en éstos parámetros, incluyendo las del Sistema Internacional. Si solamente se establece el valor de `width`, la altura se escalará proporcionalmente.

También es posible establecer la longitud de la imagen respecto a otros elementos en el documento. Imagen con la misma anchura que el texto:



Finalmente, las imágenes pueden rotarse fácilmente.

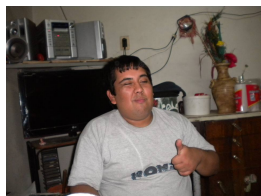


## 4.2. Posicionar imagen

Colocar las imágenes de un modo particular es requerido porque el programa a veces las inserta, y sepa el lector disculpar lo coloquial de mi prosa, en cualquier lado, o bien, porque se desea una ubicación en específico. Un ambiente puede solucionar esto. Aun así el posicionamiento de la imagen respecto del texto es quizá una de las desventajas más palpables de latex respecto de word pues requiere ser programado y uno no puede simplemente arrastrar la imagen.

Dado que este apartado incluye varios puntos importantes, lo haremos en forma de lista y le enseñó a hacer una. Cabe aclarar que existen diversos tipos de listas, enumeradas y con viñetas, que casi no difieren en su código. A continuación usaremos las enumeradas.

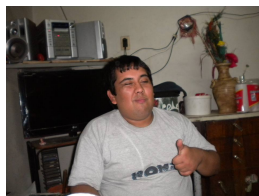
1. Comando `figure`. El ambiente `figure` se usa para mostrar las imágenes como elementos flotantes dentro del documento. Ésto significa que basta con incluir la imagen dentro del ambiente `figure` y ya no tiene que preocuparse por dónde aparecerá, latex la posicionará de tal manera que encaje con el flujo del documento. Aunque esto suena muy bien, a veces uno quiere tener control sobre dónde aparecerán las imágenes (se ve que la imagen al tope de esta página ha quedado mal ubicada pues debería ir justo debajo de este párrafo, según el código). Un parámetro adicional se puede usar para establecer el posicionamiento de la figura, los mismos se tratan seguidamente y son válidos para las tablas.
2. Parámetros importantes y centrado. Estos se insertan entre corchetes como se muestra enseguida. Se los organiza en una tabla.



Parámetro	acción
h	Aproximadamente en el mismo punto donde aparece en el código
t	Inserta la figura al inicio de la página.
b	Inserta la figura al final de la página.
p	Inserta los elementos en una página por separado, que sólo contiene figuras.

Si se agrega el paquete `float`, se obtiene un nuevo posicionamiento (H) que coloca la imagen exactamente donde aparece en el código.

Un ejemplo de lo que sucede al agregar el comando “h” al caso anterior se toma ahora. Como se ve, la figura está correctamente ubicada en esta ocasión. El comando adicional “centering” se encarga de centrar la imagen. El valor de alineación por defecto es más bien a la izquierda.

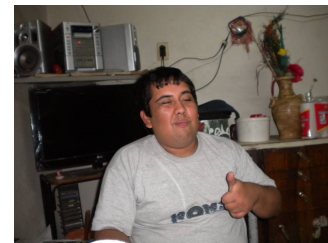


### 3. Texto rodeando imagen

Equivale al “diseño cuadrado” de word y aquí requiere ser programado aunque realmente queda muy bien. Es posible envolver el texto alrededor de la figura. Para explicar este caso comenzaremos con el ejemplo. Para que los comandos en el mismo funcionen, se debe importar el paquete `wrapfig` al inicio (en el preámbulo) “`usepackage{wrapfig}`”. Esto permite usar el ambiente `wrapfigure` mediante “`begin{wrapfigure}{r}{0.25 textwidth}`”. Tiene dos parámetros adicionales entre llaves. El primero, establece la alineación de la figura; usa `l` (por izquierda en inglés) para izquierda y `r` para derecha (por derecha en inglés). El segundo, establece la anchura de la caja que contiene a la imagen. En el ejemplo se usan unidades relativas, pero unidades absolutas (cm, mm, in, etc) pueden usarse. Si bien el comando “centering” se explicó con anterioridad, aquí la imagen será centrada respecto a la caja que la contiene.

Se aclara que si se reemplaza `enumerate` por `itemize`, se obtiene una lista con viñetas en lugar de números, pueden hacerse listas combinadas, abriendo un ambiente dentro del otro.

**Nota:** Quizá usted se esté preguntando la razón de que se cerrara el ambiente `enumerate` antes de abrir el relativo a la imagen encerrada por texto; por alguna razón latex no deja usar este comando estando activo el primero.



### 4.3. Referencias y epígrafes

El poder añadir leyendas con una breve descripción de la imagen y etiquetas que luego se puedan usar para referirse a ella son herramientas geniales cuando se trabaja con documentos muy largos. Es tan fácil, sólo hay que añadir el comando “`caption{epígrafe}`” dentro de las llaves escribir el texto a mostrar. El lugar donde se imprimirá la leyenda depende de dónde se inserta aquél (antes o después de `includegraphics`). Otro ejemplo se muestra seguidamente.

Las figuras pueden ser referidas dentro del texto llamándolas (preste atención porque esto se aplica a tablas, ecuaciones, teoremas y demás), sin necesidad de estar recordando el número de figura, es muy útil y una ventaja con respecto a otros procesadores. Esto está relacionado con el comando `label` de más arriba que establece la etiqueta para la figura. Dado que las mismas se pueden usar en diversos tipos de elementos en el documento, es una buena práctica el usar un prefijo, tal como `fig`: en este caso. Así, si quiere llamar al número de la imagen en cuestión toma 1 y este comando lo imprime. Finalmente, si se desea nombrar el número de la página en la que

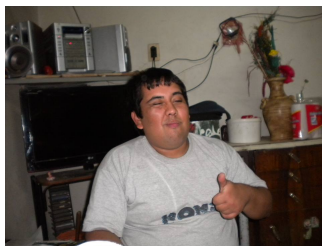


Figura 1: Captura de un gordo cool

aparece tal imagen de manera automática, puede hacerse así: el gordo está en la página 8. El comando `caption` es obligatorio para poder añadir una etiqueta y referir la figura.

## 5. Formato

Es necesario en ocasiones dar un formato específico distinto al que es dado por defecto. En este apartado se dan algunas herramientas que suele ser preciso conocer.

### 5.1. Negrita, subrayado, cursiva y énfasis

Se explora varias utilidades clásicas de todo procesador de texto, las mismas son muy fáciles de comprender. Son el subrayado, **la negrita**, *la bastardilla* y el *énfasis*. Un ejemplo sencillo basta para clarificar el caso, recuerde que es *posible* combinarlas.

### 5.2. Encabezados y pies de páginas

Son recursos necesarios y hay maneras diversas de colocarlos. La configuración que viene por defecto coloca los números de páginas en el centro y abajo, pero fácilmente podemos modificarla como se muestra seguidamente. Ello se logra trabajando en el preámbulo y dejándolo definido para todo el documento. El paquete `fancyhdr` incorpora estas funciones. El resto de los comandos se explican ahora. El `pagestyle` especifica el estilo que usaremos para los encabezados, existen diversos y en este caso se ha seleccionado uno muy tradicional, el `fancy`; otros utilizables son `empty` (que deja en blanco ambos espacios), `plain` (el que se coloca por defecto con el número de página centrado al pie) y `myheadings`, que coloca el número de página a la derecha.

El código aquí proporcionado (en el preámbulo) permite trabajar de forma más general. El comando `fancyhf` limpia los espacios en cuestión para que no aparezca el ajuste por defecto; los comandos `rhead`, `lhead`, `chead` (no se ha usado aquí el último de estos), `rfoot`, `lfoot` y `tfoot` (tampoco se usan aquí los dos últimos), no requieren demasiada explicación. Se coloca entre llaves el texto que se desea aparezca.

### 5.3. Formato del párrafo

Se tratan ahora diversas herramientas asociadas al formato del texto, es lo que en Word llamamos “Párrafo”, incluye configurar una sangría, el interlineado o el espacio entre párrafos.

#### 5.3.1. Sangría

El comando que aquí se muestra es el que gobierna la configuración de la sangría y puede cambiarse fácilmente afectando la cantidad entre llaves.

Si se va usar en el texto siempre la misma sangría conviene dejar colocado este comando en el preámbulo. En caso contrario, puede definirse justo encima de cada sección donde se desee cambiar la configuración.

También puede volverse a la configuración que aparece por defecto.

#### 5.3.2. Espacio entre párrafos

Para determinar el espacio entre un párrafo y el texto posterior se puede utilizar el comando `setlength{parskip}`.



El mismo rige para todo el texto a partir del cual el comando se coloca. Y puede volverse a la configuración por defecto.

Como se ve, todo esto que en latex necesita programarse, se puede hacer de forma interactiva en otros procesadores pero la forma es mucho más general y una vez que se aprende a usarlo evita grandes pérdidas de tiempo.

### 5.3.3. Interlineado

Si bien existen varios comandos para controlar esto, el aquí usado es principal, y permite satisfacer las necesidades básicas. El código asociado está en el preámbulo, pues el interlineado se define allí y es único para todo el cuerpo del texto; el valor definido es 1, que coincide con el que viene por defecto, pero puede modificarse.

### 5.3.4. Justificado, alineado a izquierda, alineado a derecha y centrado

- Alineado izquierda

Se debe usar el comando aquí mostrado.

- Alineado derecha

Se usa un comando análogo

- Centrado

Se auto-explica

- Justificado El comando justify permite que todo el texto subsiguiente esté justificado.

## 5.4. Uso de colores

Si bien los colores muchas veces son pocos utilizados es necesario saber cómo pueden configurarse. Como no debería sorprenderle a esta altura, para usar los colores debe invocarse un paquete asociado, el mismo está en el preámbulo. Si se desea que todo un ambiente esté coloreado puede hacerse lo siguiente.

- First item

- Second item

Si solo una **palabra** o frase desea cambiarse debe usar este comando pero si en cambio se desea **resaltar** debe usarse este otro.

Puede delimitar con llaves <sup>1</sup> los elementos coloreados, tal como se muestra en el caso de la siguiente línea<sup>2</sup> del ancho del cuerpo de texto.

## 5.5. Referenciar secciones y ecuaciones

Tal como se mostró con las imágenes se puede hacer para las secciones. Es muy útil pues cuando se agrega una nueva sección intermedia siempre está el problema de modificar todos los sitios donde se mencionan secciones por su número, para ello podemos usar el siguiente truquillo, que se muestra en la subsección 5.5.

También puede utilizarse para ecuaciones, como en el ejemplo

$$\sum_{i=0}^{\infty} a_i x^i \tag{2}$$

La ecuación 2 es una serie de potencias, que aparece en la página 9.

<sup>1</sup>Pueden colocarse notas al pie de página

<sup>2</sup>Es realmente muy sencillo y se autonumeran

## 5.6. Tamaños y tipos de letra

Si bien latex los elige adecuadamente de acuerdo a la clase de documento puede desear cambiarlos, lo que es relativamente sencillo. Existen distintos tipos de letra y se los puede afectar con negritas, subrayado, como ya se exploró. Se muestran en una tabla los más representativos. Hay más tamaños y fuentes que los que se exhiben.

Tamaño
El más pequeño
Pequeño
Grande
Más grande
Muy Grande

Con respecto a las fuentes, se muestran algunas tradicionales.

Tamaño
Serif (roman)
Sans serif
Typewriter (monospace)

Se aclara que de colocar una barra invertida y el comando con el tipo o tamaño de letra sin llave, todo el texto subsiguiente adquirirá ese formato.

## 6. Referencias Bibliográficas

En L<sup>A</sup>T<sub>E</sub>X, las mismas son un caso particular que requiere un poco de estudio extra, sin embargo, quedan realmente muy prolijas y normalizadas, cosa que los ingenieros que usamos latex, apreciamos en demasía. Primero es necesario introducir un paquete asociado (biblatex) para poder trabajar de manera prolija, luego, el comando que sigue en el preámbulo a ese (`\addbibresource{ejemplo.bib}`), es el que importa la información asociada a cada referencia desde un archivo .bib que luego crearemos. Para ello en la pantalla a la izquierda del editor debemos crear un nuevo archivo que llamaremos “ejemplo.bib”. Aparece ahora entre nuestros archivos, uno nuevo que podremos editar. Ese archivo .bib tiene una sintaxis propia que se explora en el archivo adjunto, es muy intuitiva y fácil de entender. Ve ahora al documento .bib, explóralo y lee el comentario colocado (se encuentra al final del código).

Ahora podemos llamar directamente a nuestra bibliografía desde aquí. Para que la bibliografía aparezca, esta debe ser citada en algún momento en el texto.[1].[2].[3].[4]. Se aclara que son de fantasía estas cuatro y no se utilizan realmente en este texto.

### Las Referencias fantasía

- [1] Paul Adrien Maurice Dirac. *The Principles of Quantum Mechanics*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.
- [2] Albert Einstein. “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]”. En: *Annalen der Physik* 322.10 (1905), págs. 891-921. DOI: <http://dx.doi.org/10.1002/andp.19053221004>.
- [3] Donald Knuth. *Knuth: Computers and Typesetting*. URL: <http://www-cs-faculty.stanford.edu/~uno/abcde.html>.
- [4] Donald E. Knuth. “Fundamental Algorithms”. En: Addison-Wesley, 1973. Cap. 1.2.
- [5] Share Latex. *ShareLaTeX guides*. URL: <https://es.sharelatex.com/learn>.

También pueden ser subclasificadas de la siguiente forma y colocarse automáticamente en la tabla de contenidos como se muestra para el caso de un artículo de referencia.

### Artículos fantasía

- [2] Albert Einstein. “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]”. En: *Annalen der Physik* 322.10 (1905), págs. 891-921. DOI: <http://dx.doi.org/10.1002/andp.19053221004>.

## Libro fantasía

- [1] Paul Adrien Maurice Dirac. *The Principles of Quantum Mechanics*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.

## L<sup>A</sup>T<sub>E</sub>X Fantasía

- [3] Donald Knuth. *Knuth: Computers and Typesetting*. URL: <http://www-cs-faculty.stanford.edu/~uno/abcde.html>.

[5]

## 7. Paquetes específicos útiles

### 7.1. Paquete circuitos










Por ser particularmente útil para ingeniería se explorará uno de los tantos paquetes asociados a campos de conocimiento específico en Latex, tal resulta ser el de circuitos, que permite de manera sencilla introducir diagramas circuitales de forma prolija y rápida.




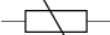


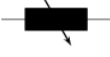







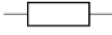

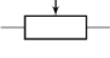
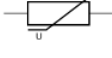
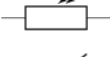











Primero es necesario importar el paquete asociado al preámbulo. El sistema es muy sencillo y se basa en definir la posición de los elementos a partir de puntos en el plano, que viene a ser su papel. Se coloca un ejemplo sencillo como motivación. El comando draw es el que permite dibujar la circuitería. El sistema de coordenadas es cartesiano, con el eje X horizontal, positivo a la derecha, y el Y vertical, positivo hacia arriba, de manera de nombrar los puntos como (x,y).



















Luego, trabajaremos con un ejemplo más complejo, para el que es necesario conocer los comandos que este paquete incluye. Los mismos se pueden ver en las siguientes imágenes [5].

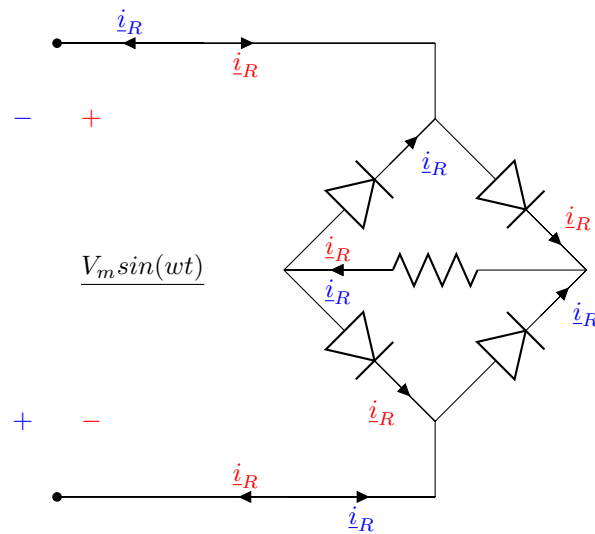
capacitor		polar capacitor		variable capacitor	
cute inductor		variable cute inductor		american inductor	
variable american inductor		europaean inductor		variable european inductor	
transmission line		vsourcesin		isourcesin	
closing switch		opening switch		push button	

ground 	sground 	nground 
rground 	pground 	cground 
tlinestub 	antenna 	rxantenna 

ammeter 	voltmeter 	short 
open 	lamp 	generic 
tgeneric 	ageneric 	fullgeneric 
tfullgeneric 	memristor 	american resistor 
vR 	american potentiometer 	european resistor 
european resistor 	european potentiometer 	varistor 
photoresistor 	thermocouple 	thermistor 
thermistor ntc 	fuse 	afuse 
battery 	battery1 	european voltage source 
american voltage source 	european current source 	american current source 

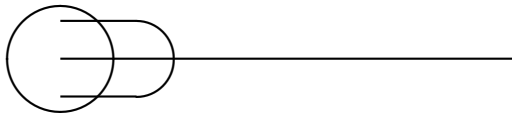
empty diode 	empty Schottky diode 	empty Zener diode 
empty tunnel diode 	photodiode 	empty diode 
empty varcap 	full diode 	full Schottky diode 
full Zener diode 	full tunnel diode 	full photodiode 
full led 	full varcap 	squid 
barrier 		

Con esta información, el ejemplo dado, lo que sabemos de expresiones matemáticas y algo de ingenio deberíamos ser capaces de construir casi cualquier circuito.

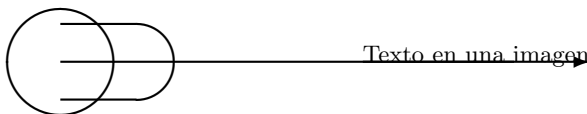


## 7.2. Diagramas simples

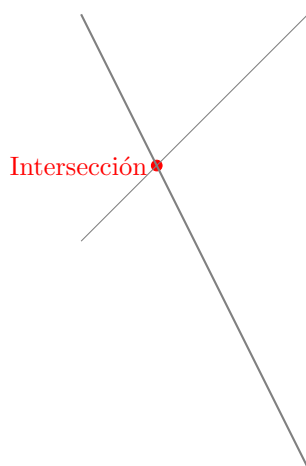
Si se desea graficar curvas, círculos, triángulos o líneas puede utilizarse un paquete especial (tikz) que se ha llamado en el preámbulo. Antes de hablar específicamente sobre él, introduciremos algunos conceptos clave relacionados con los diagramas más simples que no requieren un paquete especial, solo un ambiente particular “picture”. Primero analizaremos un caso que permite comprender el manejo básico del apartado.



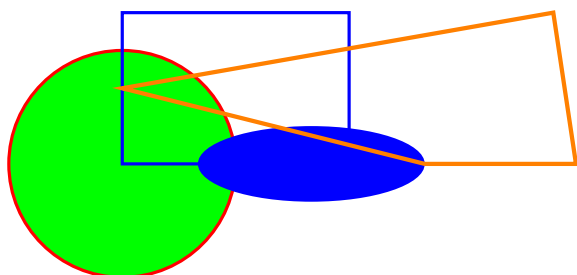
Muchas veces desean colocarse referencias o letras en las imágenes y latex también permite introducir estos elementos de manera sencilla, programando la imagen. Basta con indicar el punto sobre el que se coloca el texto, el tamaño de texto y la frase en sí, respectivamente. También pueden incluirse flechas fácilmente. El texto se inserta en el punto (6,2.2) y es de tamaño pequeño.



Una vez revisadas estas funciones básicas pasamos a trabajar con el paquete antes mencionado que es más amplio y permite funciones más complejas. Lo maravilloso de incorporarlo es que da la posibilidad de tratar a los diagramas como imágenes y posicionarlos de acuerdo a lo aprendido anteriormente para las mismas. Trae consigo capacidades como colorear elementos y rellenar circunferencias. Notar la estructura para dibujar cada elemento.

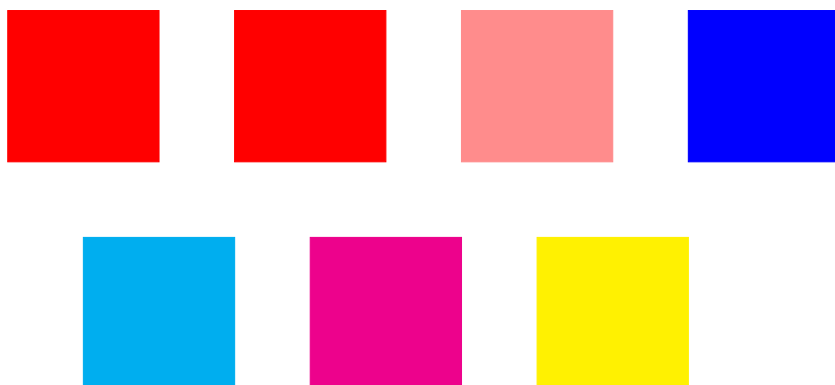


Nos ocupan ahora elementos como elipses y polígonos. Para dibujar un polígono se hace como un conjunto de líneas cerradas. Sepa que el comando `filldraw` permite crear figuras llenas donde se distingue el contorno del interior, si solo interesa el interior puede usarse `fill`. Use `draw` si tan solo dibuja elementos hechos de líneas. La estructura (una vez abierto el ambiente `tikzpicture`) empieza con el comando, se define el estilo de la línea y el relleno y luego el elemento a dibujar con sus características.

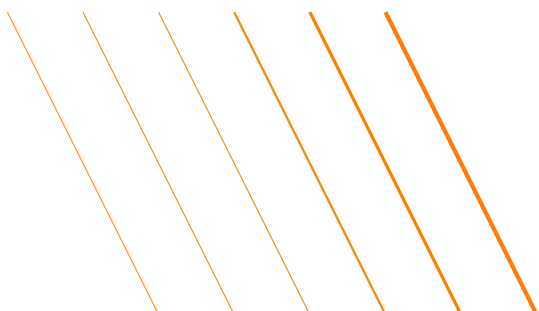


Se incorporan a continuación algunos colores y tipos de grosores de línea [5] que el paquete comprende. Cuando después de un color se coloca “!”, el mismo aparece más brillante; al colocar un número posterior se hace más claro cuanto mayor es el número.

#### ■ Colores



#### ■ Diferentes grosores de línea



Como se ve, el que las imágenes deban ser programadas las hace un poco imprácticas, pues debe definirse un sistema de coordenadas (que por defecto tiene el  $(0,0)$  abajo a la izquierda; recuerde que cuando utiliza el paquete tikz trata al diagrama como a una imanen) y la posición de los elementos en el mismo, es probablemente más conveniente generar las imágenes en otro soporte y subirlas a L<sup>A</sup>T<sub>E</sub>X.

## 8. Bibliografía

### Internet

- [5] Share Latex. *ShareLaTeX guides*. URL: <https://es.sharelatex.com/learn>.



```

\documentclass{article}
\usepackage[utf8]{inputenc} % Asociado al sistema operativo en el que se trabaja
\usepackage{geometry}
\geometry{
  a4paper,
  left=20mm,
  right=20mm,
  top=20mm,
  bottom=20mm,
} % Permite definir los márgenes y el tamaño de la hoja
\usepackage[spanish,activeacute]{babel} % Permite que el programa reconozca las ñ
y tildes.
\usepackage{biblatex}
\addbibresource{ejemplo.bib} % Se desarrolla en bibliografía
\usepackage{graphicx}
\graphicspath{ {images/} }
\usepackage{wrapfig}
\usepackage{float} % Se explica en la sección de imágenes
\usepackage{fancyhdr}
\pagestyle{fancy}
\usepackage{tabularx} % Se explica al tratar con tablas en la sección de símbolos
matemáticos
\fancyhf{}
\rhead{Tutorial Share \LaTeX}
\lhead{Gabriel De Simone}
\rfoot{Página \thepage} % Se explica al tratar con encabezados y pies de páginas
\renewcommand{\baselinestretch}{1} % Se explica en interlineado
\usepackage{xcolor} % Se explica al tratar con los colores
\usepackage{circuitikz} % Se explica en la sección de circuitos
\usepackage{tikz} % Se trata en diagramas simples
\usetikzlibrary{positioning} % Se trata en diagramas simples
\usepackage{amsthm} % Permite la incorporación del ambiente para las
demostraciones; ver teoremas y pruebas. COLOCAR ANTES QUE LOS POSTERIORES, EN CASO
CONTRARIO REDEFINE TODO LO ASOCIADO.
\newtheorem{theorem}{Teorema}
\newtheorem{corollary}{Corolario}[theorem]
\newtheorem{lemma}{Lema}[theorem] % Se explica en teoremas y pruebas
\theoremstyle{definition}
\newtheorem{definition}{Definición}[section] % Se explica en definiciones

\title{Tutorial Share \LaTeX}
\author{Gabriel De Simone }
\date{Argentina, Julio 2015}

% Aquí termina el preámbulo y comienza el cuerpo
\begin{document}

\maketitle % Permite que el título, autores y fecha se muestren en la versión PDF.
\renewcommand{\tablename}{Tabla} % El epígrafe de las tablas se nombra con la
leyenda "Tabla [número de tabla]", en lugar de "Cuadro [número de cuadro]"
\thispagestyle{empty} % La primera página aparece sin encabezado ni pie de página,
se trata en esa sección.
\tableofcontents % Agrega un índice que se actualiza automáticamente ante cualquier
cambio
\newpage % Comienza el texto siguiente en una nueva página

```

```
\begin{abstract}
```

En el presente escrito se da una breve introducción a la manera en la que latex funciona partiendo del supuesto de que el lector es un ignorante en lo respecta a esta forma de trabajar. Se muestra y explica la utilización de diversas herramientas que suelen ser de utilidad a la hora de escribir un informe. El actual desarrollo se enfoca particularmente en la escritura de artículos (entiéndase informes o escritos académicos para leer en "una sentada") aunque la inmensa mayoría de los lineamientos expuestos son válidos para otros tipos de estructuras como libros o cartas. La motivación del escrito es la no la existencia de la información aquí tratada de forma compacta y en español; la que abunda se encuentra dispersa. El lector podrá apreciar la manera de generar un artículo, al observar simultáneamente el código asociado y el resultado obtenido. No se han utilizado plantillas. % Las comillas se abren con doble tilde inversa

```
\end{abstract}
```

```
\section{Introducción}
```

`\LaTeX \space` es un procesador de texto de software, muy utilizado para la composición de literaturas académicas dado que la calidad tipográfica de los documentos obtenidos rivaliza con la de las editoriales científicas.

Permite la incorporación de secciones, índices y fórmulas de manera particularmente sencilla y ordenada, facilitando que usted se concentre en el contenido y no en el formato. Este último puede modificarse de manera general con algunas líneas de código sin tener que ir por párrafo como en otros procesadores y evita el tener que realizar constantemente ajustes de este tipo.

% El `\space` del párrafo anterior permite introducir un espacio entre el logo de latex y la palabra subsiguiente.

% Un renglón libre implica cambio de párrafo

Nos centraremos particularmente en Share `\LaTeX`, que básicamente es lo mismo pero se diferencia por permitir el trabajo online sobre un documento en forma colaborativa y en simultáneo; esto no quita que todo lo tratado sea válido para otros soportes TEX.

Antes de comenzar con la guía en sí, es de menester hacer algunas aclaraciones. Primero, el desarrollo posterior se ha basado en `\cite{sharelatexwebsite}` y una información mucho más amplia que la que aquí se expone puede ser consultada en ese sitio web. Segundo, se nombra a lo largo del texto otras bibliografías de fantasía; no se han usado en esta literatura y son solo ilustrativas. Tercero, todo lo que se muestra tiene una manera alternativa de hacerse. Cuarto, el documento busca introducir al lector de manera práctica y no se centra en cuestiones teóricas. Quinto, existen ciertos símbolos que son reservados por latex para introducir comandos, a los mismos debe anteponerles una barra derecha para imprimirlos en el PDF; algunos de estos son: `\#`, `\$`, `\%`, `\&`, `\_` y `\{`.

Para comenzar se debe tener en cuenta que el trabajo en `\LaTeX \space` se divide en tres etapas. La primera consiste en escribir el código (archivo de extensión TEX; incluye al trabajo propiamente dicho y las líneas asociadas al formato, imágenes, etc); la segunda en compilarlo, que implica pasar a PDF lo escrito y la tercera abrir el PDF. Share `\LaTeX \space` permite realizar las tres operaciones en simultáneo. Una vez en la página, cuando se crea un nuevo proyecto, se tiene la pantalla dividida; en la parte izquierda va el código y en la derecha se va mostrando el PDF a medida que recompilamos. Le recomiendo mirar en simultáneo el PDF y el TEX del presente tutorial para comprenderlo completamente, pues no se explica en el primero lo que aparece como comentario en el código, además de otros detallitos.

La parte que requiere estudio y que se trata seguidamente es la relativa al código. Todo documento trabajado en latex consta de dos secciones bien visibles en el archivo TEX. La primera es el llamado preámbulo en el que se define el tipo de texto con el que se va a trabajar, se llama a ciertos paquetes que incorporan funciones que latex no trae por defecto, se definen algunos aspectos del formato y se colocan algunos datos como el autor, la fecha; llega en el código hasta la línea `\begin{\document}`. Al igual que al programar en muchos otros lenguajes, debe indicarse el fin y el inicio de ciertas estructuras; en el caso del cuerpo del documento, que es la otra parte del código, su fin debe indicarse al terminar el

trabajo con `\end\{document\}`". Estas estructuras que comienzan y terminan de manera explícita, indicando el inicio de algún comando o función en particular, se denominan ambientes y vienen a ser semejante a las sentencias `if`, `for` o `while` de otros lenguajes en este sentido.

En el preámbulo se han colocado toda una serie de paquetes que se irán utilizando a lo largo del texto y se explicarán oportunamente. Ahora resulta oportuno tratar la primera línea de todo trabajo en latex: `documentclass\{clase de documento\}`. Esta línea define el formato básico que se utilizará. Puede colocarse artículo (`article`), libro (`book`), informe (`report`), entre tantos otros de acuerdo a las necesidades. Determina cosas como el tamaño de las sangrías, espacios entre títulos o la forma de numerar elementos según el tipo de texto, basándose en las estructuras de más uso en esos escritos. Muchos de estos valores por defecto pueden modificarse (lo haremos) e incluso usted puede crear su propio "documentclass" como se enseña en diversas bibliografías. Es necesario colocar esta línea porque le da orden a todo lo que usted no ajuste manualmente y permite tener una base desde dónde comenzar a trabajar.

Una vez que se da inicio al cuerpo del texto, puede fácilmente incorporar secciones, subsecciones y capítulos, que se numeran automáticamente y aparecen con el mismo formato según el caso. Su estudio se realiza al progresar en el documento. Se aclara desde ahora que Latex cuenta con distintos niveles y subniveles para organizar contenidos que son desde más a menos general: parte (`part`), capítulo (`chapter`), sección (`section`), subsección (`subsection`), subsubsección (`subsubsection`), párrafo (`paragraph`) y subpárrafo (`subparagraph`); todos se inician anteponiendo una barra invertida. No todos están disponibles en el formato artículo.

`\section{Expresiones matemática}`

Por ser de las principales ventajas y como a los ingenieros técnicamente nos importa la matemática, es importante poder expresarla y escribir fórmulas en latex. Veremos cómo hacerlo.

Para colocar una fórmula en línea con el texto puede usarse la siguiente estructura `$x^2 + y^2 = z^2$`. En cambio para hacerlo al estilo word, podemos hacer así: `$$x^n + y^n = z^n$$` Es el famoso teorema de Pitágoras generalizado, que como sabrá no es válido.

La ecuación debe ser incluida dentro del ambiente `equation` si se requiere que aparezca numerada. Con la siguiente estructura queda prolija y se muestra cómo colocar una fracción.

```
\begin{equation}
\begin{split}
A = \frac{\pi r^2}{2} = \frac{1}{2} \pi r^2
\end{split}
\end{equation}
```

% El ambiente `split` le indica que las expresiones separadas por el igual van en distintos miembros.

`\subsection{Símbolos matemáticos (y tablas)}`

Existen diversos y la forma de llamarlos en latex se coloca en la siguiente tabla, y de paso le enseño a poner una tabla.

`\begin{table}[h]` % Permite definir los parámetros de la tabla a crear; los mismos van entre corchetes, como la ubicación, `[h]`, que indica que se coloque aquí (here). Si se desee usar los ajustes por defecto no es necesario este ambiente. Se trata con más detalle en imágenes lo asociado a posicionamiento.

`\begin{center}` % Permite centrar cualquier elemento, en este caso, la tabla.

`\begin{tabular}{|c|c|}` % Describe la forma de la tabla. Las barras verticales indican líneas verticales de separación en la tabla, cada `c` implica una columna donde el contenido está centrado en la misma (con una `l` está a la izquierda y con una `r` a la derecha). Es la forma básica de la tabla.

`\hline` % Implica la colocación de un línea horizontal

Descripción & código `\\` % Los & separan elementos de una fila

`\hline\hline`

```

Letras griegas & \alpha \beta \gamma \rhoho \sigma \delta \epsilonpsilon \\ % La doble
barra inclinada implica final de la fila
\hline
Operadores binarios & \times \otimes \oplus \cup \cap \\
\hline
Operadores de relación & \subset \supset \subseteq \supseteq \\
\hline
Otros importantes & \int \oint \sum \prod \\
\hline
\end{tabular}
\end{center}
\caption{Tabla de símbolos} % Epígrafe de la tabla, se numera automáticamente
\label{tabla:ta} % Referencia tabla, permite llamar al número de la tabla en el
texto como se muestra con detalle en imágenes.
\end{table} % Terminar tabla
% En "posicionar imágenes" y en "referenciar secciones y ecuaciones" se explica
con más cuidado algunos detalles que también afectan a las tablas.

```

Cuando en una columna de una tabla se debe insertar un texto particularmente extenso, el ambiente `tabular` no puede resolver la situación de manera conveniente y la tabla escapa al ancho de la página. Un ambiente que soluciona el caso es el llamado `tabularx`, que se utiliza de la misma manera, salvo que en las columnas afectadas por estas longitudes de texto infrecuentemente grandes, debe colocarse una `X`, `\space` al declarar la estructura básica de la tabla. Un ejemplo se muestra a continuación. Debe incorporarse en el preámbulo el paquete `\usepackage{tabularx}` pues este ambiente no lo trae latex por defecto.

Debe tenerse en cuenta que este ambiente no es compatible con el comando `caption`, por lo que aquí se muestra una forma de colocar un epígrafe a estas tablas. La trampa consiste en colocar un ambiente `tabular` después de la tabla, sin una tabla dentro; se muestra seguidamente.

```

\begin{center} % Permite centrar cualquier elemento, en este caso, la tabla.

```

```

\begin{tabularx}{\textwidth}{|c|c|c|X|X|} % Describe la forma de la tabla. Las
barras verticales indican líneas verticales de separación en la tabla, cada c
implica una columna donde el contenido está centrado en la misma (con una l está a
la izquierda y con una r a la derecha). Es la forma básica de la tabla.

```

```

\hline
Nombre & Símbolo & Activo/pasivo & Características & Aplicaciones \\
\hline\hline
Resistor & \begin{circuitikz} \draw
(0,0) to[american resistor] (2,0); %Se explica al tratar con circuitos
\end{circuitikz} & Pasivo & Se opone al paso de corriente, generando una caída de
tensión que se traduce en potencia disipada.
Existen de materiales diversos que constan de resistividad variable con la
temperatura y con la que la resistencia varía proporcionalmente, dependiendo
también de la geometría. Cumplen la ley de Ohm. Su efecto se cuantifica en Ohmios.
& Contro de volumen y tono en los dispositivos de sonido, control de brillo, etc.
Existe un tipo de potenciómetro que se fabrica especialmente para ajustar voltajes
o corrientes en algunos circuitos integrados. \\
\hline
\end{tabularx}

\begin{table}[H]
\caption{Componentes de circuito}
\label{tabla:ta}
\end{table}

```

`\end{center}`

`\subsection{Subíndices y superíndices}`

Usted podría pensar que son algo secundarios, estando equivocado, pues realmente los usamos a menudo y hacerlos con word es realmente engorroso. Los mismos se escriben tan solo anteponiendo un símbolo al índice tal como veremos enseguida.

Por ejemplo, para una integral definida podemos hacer lo siguiente `$$\int\limits_0^1 x^2 \, dx` Se ve claramente que el "guión" `\space` coloca el subíndice y el "piquito (diría una vieja)" `\space` el superíndice. Si los mismos son largos puede hacerse de la siguiente manera que es muy cómoda. `$$ x^{\alpha} - 1 = y_{ij} + y_{ij}` Este último que le pongo ahora es realmente excelente porque las sumatorias se usan bastante y las fracciones más todavía, también aparece el símbolo del infinito. `$$\sum_{i=1}^{\infty} \frac{1}{n^s} = \prod_p \frac{1}{1 - p^{-s}}` `$$`

`\subsection{Corchetes y paréntesis}`

Es la forma de colocar las matrices y los vectores, que tan poco usamos a decir verdad, pero vale la pena ponerlo. Los mismos pueden ajustarse de manera automática o manual. Primero la automática, que se basa en construir una tabla como la de antes y colocarle el símbolo en cuestión a la derecha y la izquierda.

```

$$
\left \{ % Indica que se coloque una llave a la izquierda cuyo tamaño se ajusta
automáticamente
\begin{tabular}{ccc} % La tabla sigue la lógica de la anterior
1 & 5 & 8 \\
0 & 2 & 4 \\
3 & 3 & -8
\end{tabular}
\right \} % Indica que se coloque una llave a la derecha cuyo tamaño se ajusta
automáticamente

```

En forma manual se deben colocar los siguientes símbolos que se presentan en la tabla `\cite{sharelatexwebsite}`

```

\begin{center}
\begin{tabular}{|c|c|} % Las barras verticales indican líneas verticales de
separación en la tabla, cada c implica una columna donde el contenido está centrado
en la misma.
\hline
Descripción & código \\
\hline\hline
Paréntesis & \big( \Big( \bigg( \Bigg( \\
\hline
Corchetes & \big] \Big] \bigg] \Bigg] \\
\hline
Llaves & \big\{ \Big\{ \bigg\{ \Bigg\{ \\
\hline
\end{tabular}
\end{center}
Y para cerrar un ejemplo clásico $$\Big( 90+76z \Big)$$

```

`\subsection{Teoremas y pruebas}`

Para dar un formato particular a estos, debemos primero definir cómo los llamaremos y la palabra que aparecerá en el PDF cuando se lo haga, lo mismo se hace en el preámbulo a través del comando `\newtheorem{theorem}{Teorema}`, que tiene dos parámetros. El primero es el nombre del ambiente definido (algunos ambientes es necesario definirlos) y el segundo, el nombre con el que aparece en el texto. El

[theorem] que aparece en las dos líneas siguientes a esa, indica que el contador de los corolarios y lemas debe reiniciarse al iniciar un nuevo teorema; si en cambio, se quiere que usen el mismo que los teoremas, debe colocarse antes del último de los dos textos entre llaves en esas líneas y si se quiere que tengan su propia numeración no debe colocarse.

```
\begin{theorem}
Dios todo lo puede.
\end{theorem}
```

```
\begin{theorem}[Teorema generalizado] % El texto entre corchetes aparecerá como
nombre del teorema
\label{tg} % Permite luego referenciar el teorema
Es la generalización del teorema anterior.
Dios todo lo puede, cuando quiere y como quiere.
\end{theorem}
```

Las consecuencias del teorema `\ref{tg}` se ven en la realidad que lo rodea.

```
\begin{corollary}
No desafíe a Dios.
\end{corollary}
```

```
\begin{lemma}
Dadas dos personas, harán ellas lo que Dios desee, no su libertad.
\end{lemma}
\begin{proof} % Permite colocar formato de demostración
No necesito más prueba que el antiguo testamento.
\end{proof}
```

```
\subsection{Definiciones}
```

Una manera prolija de dar definiciones es usando el formato de los teoremas que realmente queda maravilloso. El paquete `amsthm` permite introducir esta función, `\textbf{y}` también el formato anterior para demostraciones}. Los otros comandos `theoremstyle\{definition\}` y

`newtheorem\{definition\}\{Definición\}` [section], permiten crear un ambiente (definition) con el estilo de los teoremas, darle un nombre con el que aparecerá en el texto y reiniciar su contador en cada sección.

```
\begin{definition}{Chiripioca}
Dícese del estado de una persona iracunda y desencajada. Ej: Le agarró la...
\end{definition}
```

```
\section{Operadores}
```

Los operadores tienen una notación particular en latex. Se destacan algunos.

```
\subsection{Límites}
```

Para colocarlos tan prolijo como en el `stuart` o en el `eduardts` tan solo se hace lo siguiente, se considera que el código se explica a sí mismo.

```
$$
\lim_{h \rightarrow 0} \frac{f(x+h)-f(x)}{h} = \frac{df(x)}{dx}
$$
```

que queda de esta forma si se usa en línea con el texto `$ \lim_{x \rightarrow h} (x-h) $`.

```
\subsection{Operadores de referencia}
```

Para tenerlos a mano, se adjuntan en la tabla siguiente.

```

\begin{center}
\begin{tabular}{|c|c|}
\hline
Descripción & código \\
\hline\hline
Seno & \sin \\
\hline
Arcoseno & \arcsin \\
\hline
Seno hiperbólico & \sinh \\
\hline
\end{tabular}
\end{center}

```

Son bastante intuitivos y los demás se toman de modos semejantes.

### \section{Imágenes}

Un buen trabajo pasa a ser excelente cuando se agrega la cantidad de imágenes correcta de adecuada forma, prolijamente, con el tamaño adecuado y sus epígrafes. Esto nos ocupa ahora mismo.

#### \subsection{Insertar imágenes, cambiarles el tamaño y rotarlas}

Latex no tiene la capacidad de manipular imágenes por sí mismo, por esa razón se debe importar el paquete `graphicx` en el preámbulo. Para usar dicho paquete se añadió al inicio la siguiente línea `\usepackage{graphicx}`. El comando `\graphicspath{{\images/}\}` le indica a latex que las imágenes están guardadas en una carpeta llamada `images` bajo el directorio actual, las mismas deben subirse a `Sharelatex` para trabajar, esto se hace con el ícono de la parte superior izquierda de la pantalla.

El comando `\includegraphics{gordo}` es el que se encarga de incluir la imagen en el documento. El nombre del archivo de imagen no debe contener espacios ni caracteres especiales.

```

\begin{center}
\includegraphics[scale=0.2]{gordo} % scale permite escalar el tamaño de la imagen
\end{center}

```

El tamaño puede modificarse con el comando `scale`. También es posible definir de forma arbitraria el ancho y el alto.

```

\begin{center}
\includegraphics[width=3cm, height=3cm]{gordo}
\end{center}

```

Como probablemente ya ha adivinado, los parámetros dentro de los corchetes `[width=3cm, height=4cm]` definen el ancho y alto de la imagen. Pueden usarse diferentes unidades de medida en éstos parámetros, incluyendo las del Sistema Internacional. Si solamente se establece el valor de `width`, la altura se escalará proporcionalmente.

También es posible establecer la longitud de la imagen respecto a otros elementos en el documento. Imagen con la misma anchura que el texto:

```

\includegraphics[width=\textwidth, height=4cm]{gordo}

```

Finalmente, las imágenes pueden rotarse fácilmente.



```
\begin{center}
\includegraphics[scale=0.1, angle=67]{gordo} %grados sexagesimales, los de toda la
vida
\end{center}
```

```
\subsection{Posicionar imagen}
```

Colocar las imágenes de un modo particular es requerido porque el programa a veces las inserta, y sepa el lector disculpar lo coloquial de mi prosa, en cualquier lado, o bien, porque se desea una ubicación en específico. Un ambiente puede solucionar esto. Aun así el posicionamiento de la imagen respecto del texto es quizá una de las desventajas más palpables de latex respecto de word pues requiere ser programado y uno no puede simplemente arrastrar la imagen.

Dado que este apartado incluye varios puntos importantes, lo haremos en forma de lista y le enseño a hacer una. Cabe aclarar que existen diversos tipos de listas, enumeradas y con viñetas, que casi no difieren en su código. A continuación usaremos las enumeradas.

```
\begin{enumerate}
```

```
\item Comando figure.
```

El ambiente figure se usa para mostrar las imágenes como elementos flotantes dentro del documento. Esto significa que basta con incluir la imagen dentro del ambiente figure y ya no tiene que preocuparse por dónde aparecerá, latex la posicionará de tal manera que encaje con el flujo del documento. Aunque esto suena muy bien, a veces uno quiere tener control sobre dónde aparecerán las imágenes (se ve que la imagen al tope de esta página ha quedado mal ubicada pues debería ir justo debajo de este párrafo, según el código). Un parámetro adicional se puede usar para establecer el posicionamiento de la figura, los mismos se tratan seguidamente y son válidos para las tablas.

```
\begin{figure}
```

```
\includegraphics[scale=0.1]{gordo}
```

```
\end{figure}
```

```
\item Parámetros importantes y centrado.
```

Estos se insertan entre corchetes como se muestra enseguida.

Se los organiza en una tabla.

```
\begin{center}
\begin{tabular}{||c | c||}
\hline
Parámetro & acción \\\
\hline\hline
h & Aproximadamente en el mismo punto donde aparece en el código \\\
\hline
t & Inserta la figura al inicio de la página. \\\
\hline
b & Inserta la figura al final de la página. \\\
\hline
p & Inserta los elementos en una página por separado, que sólo contiene figuras.
\\
\hline
\end{tabular}
\end{center}
```

Si se agrega el paquete float, se obtiene un nuevo posicionamiento (H) que coloca la imagen exactamente donde aparece en el código.



Un ejemplo de lo que sucede al agregar el comando ```h" \space` al caso anterior se toma ahora. Como se ve, la figura está correctamente ubicada en esta ocasión. El comando adicional ```centering` se encarga de centrar la imagen. El valor de alineación por defecto es más bien a la izquierda.

```
\begin{figure}[h]
\includegraphics[scale=0.1]{gordo}
\centering
\end{figure}
\item Texto rodeando imagen

\end{enumerate}

\begin{wrapfigure}{r}{0.25\textwidth} % La figura estará a la derecha
\centering
\includegraphics[width=0.25\textwidth]{gordo}
\end{wrapfigure}
```

Equivale al ```diseño cuadrado` de word y aquí requiere ser programado aunque realmente queda muy bien. Es posible envolver el texto alrededor de la figura. Para explicar este caso comenzaremos con el ejemplo.

Para que los comandos en el mismo funcionen, se debe importar el paquete `wrapfig` al inicio (en el preámbulo) ```usepackage{wrapfig}`. Esto permite usar el ambiente `wrapfigure` mediante ```begin{\wrapfigure}{r}{0.25\textwidth}`. Tiene dos parámetros adicionales entre llaves. El primero, establece la alineación de la figura; usa `l` (por izquierda en inglés) para izquierda y `r` para derecha (por derecha en inglés). El segundo, establece la anchura de la caja que contiene a la imagen. En el ejemplo se usan unidades relativas, pero unidades absolutas (cm, mm, in, etc) pueden usarse. Si bien el comando ```centering` se explicó con anterioridad, aquí la imagen será centrada respecto a la caja que la contiene.

Se aclara que si se reemplaza `enumerate` por `itemize`, se obtiene una lista con viñetas en lugar de números, pueden hacerse listas combinadas, abriendo un ambiente dentro del otro.

**\textbf{Nota}**: Quizá usted se esté preguntando la razón de que se cerrara el ambiente `enumerate` antes de abrir el relativo a la imagen encerrada por texto; por alguna razón latex no deja usar este comando estando activo el primero.

**\subsection{Referencias y epígrafes}**

El poder añadir leyendas con una breve descripción de la imagen y etiquetas que luego se puedan usar para referirse a ella son herramientas geniales cuando se trabaja con documentos muy largos. Es tan fácil, sólo hay que añadir el comando ```caption{\epígrafe}` y dentro de las llaves escribir el texto a mostrar. El lugar donde se imprimirá la leyenda depende de dónde se inserta aquél (antes o después de `includegraphics`). Otro ejemplo se muestra seguidamente.

```
\begin{figure}[h]
\centering
\includegraphics[width=0.25\textwidth]{gordo}
\caption{Captura de un gordo cool}
\label{fig:gordo} % Ver párrafo siguiente
\end{figure}
```

Las figuras pueden ser referidas dentro del texto llamándolas (preste atención porque esto se aplica a tablas, ecuaciones, teoremas y demás), sin necesidad de estar recordando el número de figura, es muy útil y una ventaja con respecto a otros procesadores. Esto está relacionado con el comando `label` de más arriba que establece la etiqueta para la figura. Dado que las mismas se pueden usar en diversos tipos de elementos en el documento, es una buena práctica el usar un prefijo, tal como `fig`: en este caso. Así, si quiere llamar al número de la imagen en cuestión toma `\ref{fig:gordo}` y este comando lo imprime. Finalmente, si se desea nombrar el número de la página en la que aparece tal imagen de manera automática, puede hacerse

así: el gordo está en la página `\pageref{fig:gordo}`. El comando `caption` es obligatorio para poder añadir una etiqueta y referir la figura.

### `\section{Formato}`

Es necesario en ocasiones dar un formato específico distinto al que es dado por defecto. En este apartado se dan algunas herramientas que suele ser preciso conocer.

#### `\subsection{Negrita, subrayado, cursiva y énfasis}`

Se explora varias utilidades clásicas de todo procesador de texto, las mismas son muy fáciles de comprender. Son `\underline{}` el subrayado, `\textbf{}` la negrita, `\textit{}` la bastardilla y el `\emph{}` énfasis. Un ejemplo sencillo basta para clarificar el caso, recuerde `\underline{ \textbf{que es \emph{posible} combinarlas}}`.

#### `\subsection{Encabezados y pies de páginas}`

Son recursos necesarios y hay maneras diversas de colocarlos. La configuración que viene por defecto coloca los números de páginas en el centro y abajo, pero fácilmente podemos modificarla como se muestra seguidamente. Ello se logra trabajando en el preámbulo y dejándolo definido para todo el documento. El paquete `fancyhdr` incorpora estas funciones. El resto de los comandos se explican ahora. El `pagestyle` especifica el estilo que usaremos para los encabezados, existen diversos y en este caso se ha seleccionado uno muy tradicional, el `fancy`; otros utilizables son `empty` (que deja en blanco ambos espacios), `plain` (el que se coloca por defecto con el número de página centrado al pie) y `myheadings`, que coloca el número de página a la derecha.

El código aquí proporcionado (en el preámbulo) permite trabajar de forma más general. El comando `fancyhf{}` limpia los espacios en cuestión para que no aparezca el ajuste por defecto; los comandos `rhead`, `lhead`, `chead` (no se ha usado aquí el último de estos), `rfoot`, `lfoot` y `cfoot` (tampoco se usan aquí los dos últimos), no requieren demasiada explicación. Se coloca entre llaves el texto que se desea aparezca.

#### `\subsection{Formato del párrafo}`

Se tratan ahora diversas herramientas asociadas al formato del texto, es lo que en word llamamos "Párrafo", incluye configurar una sangría, el interlineado o el espacio entre párrafos.

#### `\subsubsection{Sangría}`

##### `\setlength{\parindent}{0pt}`

El comando que aquí se muestra es el que gobierna la configuración de la sangría y puede cambiarse fácilmente afectando la cantidad entre llaves.

##### `\setlength{\parindent}{25pt}`

Si se va usar en el texto siempre la misma sangría conviene dejar colocado este comando en el preámbulo. En caso contrario, puede definirse justo encima de cada sección donde se desee cambiar la configuración.

##### `\setlength{\parindent}{15pt}`

También puede volverse a la configuración que aparece por defecto.

#### `\subsubsection{Espacio entre párrafos}`

##### `\setlength{\parskip}{10mm}`

Para determinar el espacio entre un párrafo y el texto posterior se puede utilizar el comando `\setlength{\parskip}`.

El mismo rige para todo el texto a partir del cual el comando se coloca.

##### `\setlength{\parskip}{1pt}`

Y puede volverse a la configuración por defecto.

Como se ve, todo esto que en latex necesita programarse, se puede hacer de forma interactiva en otros procesadores pero la forma es mucho más general y una vez que se aprende a usarlo evita grandes pérdidas de tiempo.

`\subsubsection{Interlineado}`

Si bien existen varios comandos para controlar esto, el aquí usado es principal, y permite satisfacer las necesidades básicas. El código asociado está en el preámbulo, pues el interlineado se define allí y es único para todo el cuerpo del texto; el valor definido es 1, que coincide con el que viene por defecto, pero puede modificarse.

```
\subsubsection{Justificado, alineado a izquierda, alineado a derecha y centrado}
\begin{itemize}
\item Alineado izquierda
\begin{flushleft}
Se debe usar el comando aquí mostrado.
\end{flushleft}
\item Alineado derecha
\begin{flushright}
Se usa un comando análogo
\end{flushright}
\item Centrado
\begin{center}
Se auto-explica
\end{center}
\item Justificado
\justify
El comando justify permite que todo el texto subsiguiente esté justificado.
\end{itemize}
```

`\subsection{Uso de colores}`

Si bien los colores muchas veces son pocos utilizados es necesario saber cómo pueden configurarse. Como no debería sorprenderle a esta altura, para usar los colores debe invocarse un paquete asociado, el mismo está en el preámbulo. Si se desea que todo un ambiente esté coloreado puede hacerse lo siguiente.

```
\begin{itemize}
\color{green}
\item Firts item
\item Second item
\end{itemize}
```

Si solo una `\textcolor{red}{palabra}` o frase desea cambiarse debe usar este comando pero si en cambio se desea `\colorbox{yellow}{resaltar}` debe usarse este otro.

Puede delimitar con llaves `\footnote{Pueden colocarse notas al pie de página}` los elementos coloreados, tal como se muestra en el caso de la siguiente línea `\footnote{Es realmente muy sencillo y se autonumeran}` del ancho del cuerpo de texto.

```
\noindent % Quita la sangría
{\color{blue} \rule{\linewidth}{0.5mm} }
\indent % Restaura la sangría
\subsection{Referenciar secciones y ecuaciones} \label{referencia}
```

Tal como se mostró con las imágenes se puede hacer para las secciones. Es muy útil pues cuando se agrega una nueva sección intermedia siempre está el problema de modificar todos los sitios donde se mencionan secciones por su número, para ello

podemos usar el siguiente truquillo, que se muestra en la subsección `\ref{referencia}`.

También puede utilizarse para ecuaciones, como en el ejemplo

```
\begin{equation} \label{ec:1}
\sum_{i=0}^{\infty} a_i x^i
\end{equation}
```

La ecuación `\ref{ec:1}` es una serie de potencias, que aparece en la página `\pageref{ec:1}`.

`\subsection{Tamaños y tipos de letra}`

Si bien latex los elige adecuadamente de acuerdo a la clase de documento puede desear cambiarlos, lo que es relativamente `\huge` sencillo. Existen distintos tipos de letra y se los puede afectar con negritas, subrayado, como ya se exploró. Se muestran en una tabla los más representativos. Hay más tamaños y fuentes que los que se exhiben.

```
\begin{center}
\begin{tabular}{||c||}
\hline
Tamaño \\\
\hline\hline
{\tiny El más pequeño} \\\
{\small Pequeño} \\\
\hline
{\Large Grande} \\\
\hline
{\huge Más grande} \\\
\hline
{\Huge Muy Grande} \\\
\hline
\end{tabular}
\end{center}
```

Con respecto a las fuentes, se muestran algunas tradicionales.

```
\begin{center}
\begin{tabular}{||c||}
\hline
Tamaño \\\
\hline\hline
\texttt{Serif (roman)} \\\
\hline
\textrm{Sans serif} \\\
\hline
\textsf{Typewriter (monospace)} \\\
\hline
\end{tabular}
\end{center}
```

Se aclara que de colocar una barra invertida y el comando con el tipo o tamaño de letra sin llave, todo el texto subsiguiente adquirirá ese formato.

`\section{Referencias Bibliográficas}`

En `\LaTeX`, las mismas son un caso particular que requiere un poco de estudio extra, sin embargo, quedan realmente muy prolijas y normalizadas, cosa que los ingenieros

que usamos latex, apreciamos en demasía. Primero es necesario introducir un paquete asociado (biblatex) para poder trabajar de manera prolija, luego, el comando que sigue en el preámbulo a ese (addbibresource\{ejemplo.bib\}), es el que importa la información asociada a cada referencia desde un archivo .bib que luego crearemos. Para ello en la pantalla a la izquierda del editor debemos crear un nuevo archivo que llamaremos "ejemplo.bib". Aparece ahora entre nuestros archivos, uno nuevo que podremos editar. Ese archivo .bib tiene una sintaxis propia que se explora en el archivo adjunto, es muy intuitiva y fácil de entender. Ve ahora al documento .bib, explóralo y lee el comentario colocado.

Ahora podemos llamar directamente a nuestra bibliografía desde aquí. Para que la bibliografía aparezca, esta debe ser citada en algún momento en el texto. `\cite{dirac}.\cite{einstein}.\cite{knuthwebsite}.\cite{knuth-fa}`. Se aclara que son de fantasía estas cuatro y no se utilizan realmente en este texto.

```
% El comando heading permite añadirla a la tabla de contenidos o índice.
\printbibliography[
heading=bibintoc,
title={Las Referencias fantasía}
]
```

También pueden ser subclasificadas de la siguiente forma y colocarse automáticamente en la tabla de contenidos como se muestra para el caso de un artículo de referencia.

```
\printbibliography[heading=subbibintoc,type=article,title={Artículos fantasía}]
\printbibliography[type=book,title={Libro fantasía}]
```

```
\printbibliography[keyword={latex},title={\LaTeX \space Fantasía}]
\cite{sharelatexwebsite}
```

```
\section{Paquetes específicos útiles}
\subsection{Paquete circuitos}
```

Por ser particularmente útil para ingeniería se explorará uno de los tantos paquetes asociados a campos de conocimiento específico en Latex, tal resulta ser el de circuitos, que permite de manera sencilla introducir diagramas circuitales de forma prolija y rápida.

Primero es necesario importar el paquete asociado al preámbulo. El sistema es muy sencillo y se basa en definir la posición de los elementos a partir de puntos en el plano, que viene a ser su papel. Se coloca un ejemplo sencillo como motivación. El comando draw es el que permite dibujar la circuitería. El sistema de coordenadas es cartesiano, con el eje X horizontal, positivo a la derecha, y el Y vertical, positivo hacia arriba, de manera de nombrar los puntos como (x,y).

```
\begin{center}
\begin{circuitikz} \draw
(0,0) to[ variable cute inductor ] (2,0); % El (0,0) esta abajo a la izquierda, es
el origen de coordenadas.
\end{circuitikz}
\end{center}
```

Luego, trabajaremos con un ejemplo más complejo, para el que es necesario conocer los comandos que este paquete incluye. Los mismos se pueden ver en las siguientes imágenes `\cite{sharelatexwebsite}`.

```
\begin{figure}[h]
\centering
\includegraphics[scale=0.4]{Dynamical}
\end{figure}
\begin{figure}[h]
\centering
```

```

\includegraphics[scale=0.4]{Monopoles}
\end{figure}
\begin{figure}[H]
\centering
\includegraphics[scale=0.45]{Bipoles}
\end{figure}
\begin{figure}[H]
\centering
\includegraphics[scale=0.4]{Diodos}
\end{figure}

\newpage % Permite seguir en una página en blanco

```

Con esta información, el ejemplo dado, lo que sabemos de expresiones matemáticas y algo de ingenio deberíamos ser capaces de construir casi cualquier circuito.

```

\begin{center}
\begin{circuitikz}
\draw
(5,0) to [short, *-i_=$\color{red}\underline{i}_R$] (0,0)% Línea que va de (5,0)
al (0,0), el -* indica que hay un nodo seguido de la línea. i_=: se obtiene una
flecha en el sentido de la línea. Puede modificarse el texto que se coloca en
leyenda que acompaña a la flecha, el mismo está entre $$ .
(5,0) to [short] (5,1)
to [empty diode] (7,3)% Indica que del punto (5,1) al (7,3) hay un diodo.
to [american resistor, i_=$\color{red}\underline{i}_R$] (3,3)
to [empty diode] (5,5)
to [empty diode,i_=$\color{red}\underline{i}_R$] (7,3)
(3,3) to [empty diode,i_=$\color{red}\underline{i}_R$] (5,1)
(0,6) to [short, *-i_=$\color{red}\underline{i}_R$] (5,6)
to [short] (5,5)

(0,0) to [open, l_=$\color{red}-$] (0,2) %Indica que entre los puntos (0,0) y (0,2)
el circuito debe permanecer abierto. La leyenda se coloca con el formato l_=.
(0,0) to [open, l_=$\color{blue}+$] (0,2) % El guión en el formato i= y en l= permite
gobernar de qué lado del elemento aparece la leyenda
(0,2) to [open, l_=$\underline{v}_m \sin(\omega t)$] (0,4)
(0,4) to [open, l_=$\color{red}+$] (0,6)
(0,4) to [open, l_=$\color{blue}-$] (0,6)

(7,3)
to [open, i_=$\color{blue}\underline{i}_R$] (3,3)
(0,0) to [open,i_=$\color{blue}\underline{i}_R$] (5,0)
(3,3) to [open,i_=$\color{blue}\underline{i}_R$] (5,5)
(5,1) to [open,i_=$\color{blue}\underline{i}_R$] (7,3)
(5,6) to [open,i_=$\color{blue}\underline{i}_R$] (0,6)
\end{circuitikz}
\end{center}

```

#### \subsection{Diagramas simples}

Si se desea graficar curvas, círculos, triángulos o líneas puede utilizarse un paquete especial (tikz) que se ha llamado en el preámbulo. Antes de hablar específicamente sobre él, introduciremos algunos conceptos clave relacionados con los diagramas más simples que no requieren un paquete especial, solo un ambiente particular `\picture`. Primero analizaremos un caso que permite comprender el manejo básico del apartado.

`\bigskip` % Deja un espacio "grande" en blanco entre el texto y el diagrama, también existe el comando "`\medskip`", para un espacio mediano.

`\setlength{\unitlength}{1cm}` % Este comando establece como unidad de medida fundamental un centímetro Y NO SOLO PARA DIAGRAMAS

`\thicklines` % Gobierna el espesor de las líneas, existen otros comandos equivalentes con otros efectos sobre las líneas que más adelante se muestran; las hace un poco más gruesas.

`\begin{picture}(4,3)` % Los valores entre paréntesis corresponden al ancho y al alto de la imagen respectivamente.

`\put(2,2.2){\line(1,0){6}}` % (2,2.2) indica el punto de inicio de la línea. El (1,0) la dirección de la línea (viene a ser la pendiente) y el {numero entre llaves} la cantidad de unidades fundamentales

`\put(2,2.2){\circle{2}}` % Indica el centro del círculo y el radio

`\put(2,2.2){\oval(3,1)[r]}` % Indica un óvalo centrado en 2,2.2 y de dimensiones 3 de ancho por 1 de alto. La [r] implica que se coloca tan solo la parte derecha del óvalo, puede cambiarse por l (izquierda), b (inferior) o t (superior).

`\end{picture}`

Muchas veces desean colocarse referencias o letras en las imágenes y latex también permite introducir estos elementos de manera sencilla, programando la imagen. Basta con indicar el punto sobre el que se coloca el texto, el tamaño de texto y la frase en sí, respectivamente. También pueden incluirse flechas fácilmente. El texto se inserta en el punto (6,2.2) y es de tamaño pequeño.

`\begin{picture}(4,3)`

`\put(6,2.2){\small Texto en una imagen}`

`\put(6,2.2){\vector(1,0){3}}` %indica el punto de inicio del vector, su largo entre llaves y su dirección entre paréntesis.

`\put(2,2.2){\line(1,0){6}}`

`\put(2,2.2){\circle{2}}`

`\put(2,2.2){\oval(3,1)[r]}`

`\end{picture}`

Una vez revisadas estas funciones básicas pasamos a trabajar con el paquete antes mencionado que es más amplio y permite funciones más complejas. Lo maravilloso de incorporarlo es que da la posibilidad de tratar a los diagramas como imágenes y posicionarlos de acuerdo a lo aprendido anteriormente para las mismas. Trae consigo capacidades como colorear elementos y rellenar circunferencias. Notar la estructura para dibujar cada elemento.

`\begin{figure}[H]`

`\centering`

`\begin{tikzpicture}` % Ambiente particular incorporado por el paquete

`\filldraw[red] (0,0) circle (2pt) node[anchor=east] {Intersección};` % Crea un punto con centro en (0,0) y radio de 2pt, el mismo está relleno de color rojo. Node crea una caja donde contener el texto entre llaves, anclada a derecha o izquierda según se coloca west o east.

`\draw[gray, thick] (-1,2) -- (2,-4);` %Indica color y grosor de la línea. Se define por dos puntos en lugar de por un punto y la pendiente.

`\draw[gray, thin] (-1,-1) -- (2,2);` % Indica color y grosor de línea

`\end{tikzpicture}`

`\end{figure}`

Nos ocupan ahora elementos como elipses y polígonos. Para dibujar un polígono se hace como un conjunto de líneas cerradas. Sepa que el comando `filldraw` permite crear figuras llenas donde se distingue el contorno del interior, si solo interesa el interior puede usarse `fill`. Use `draw` si tan solo dibuja elementos hechos de líneas. La estructura (una vez abierto el ambiente `tikzpicture`) empieza con el comando, se define el estilo de la línea y el relleno y luego el elemento a dibujar con sus características.

```
\begin{figure}[h]
\begin{tikzpicture}
\filldraw[color=red!, fill=green!, very thick](0,0) circle (1.5); % Indica color
de frontera, de interior, grosor de línea, centro y radio del círculo.
\fill[blue!] (2.5,0) ellipse (1.5 and 0.5); % Indica color, posición del centro,
tipo de elemento (elipse) y el radio mayor y menor respectivamente
\draw[blue, very thick] (0,0) rectangle (3,2); % Dibuja un rectángulo. Se introduce
el punto donde el "lápiz" comienza a dibujar (0,0) y el opuesto (3,2) (sistema como
el de Ansys).
\draw[orange, ultra thick] (4,0) -- (6,0) -- (5.7,2) -- (0,1) -- cycle; % Es un
polígono, el comando cycle indica que el punto final debe unirse con el inicial.
\end{tikzpicture}
\end{figure}
```

Se incorporan a continuación algunos colores y tipos de grosores de línea `\cite{sharelatexwebsite}` que el paquete comprende. Cuando después de un color se coloca `!!`, el mismo aparece más brillante; al colocar un número posterior se hace más claro cuanto mayor es el número.

```
\begin{itemize}
\item Colores

\begin{figure}[H]
\begin{tikzpicture}
\filldraw[red!] (0,0) rectangle (2,2);
\filldraw[red] (3,0) rectangle (5,2);
\filldraw[red!45] (6,0) rectangle (8,2);
\filldraw[blue] (9,0) rectangle (11,2);
\filldraw[cyan] (1,-1) rectangle (3,-3);
\filldraw[magenta] (4,-1) rectangle (6,-3);
\filldraw[yellow] (7,-1) rectangle (9,-3);
\end{tikzpicture}
\end{figure}

\item Diferentes grosores de línea

\begin{figure}[H]
\begin{tikzpicture}
\draw[orange, ultra thin] (-1,2) -- (1,-2);
\draw[orange, very thin] (0,2) -- (2,-2);
\draw[orange, thin] (1,2) -- (3,-2);
\draw[orange, thick] (2,2) -- (4,-2);
\draw[orange, very thick] (3,2) -- (5,-2);
\draw[orange, ultra thick] (4,2) -- (6,-2);
\end{tikzpicture}
\end{figure}
\end{itemize}
```

Como se ve, el que las imágenes deban ser programadas las hace un poco imprácticas, pues debe definirse un sistema de coordenadas (que por defecto tiene el (0,0) abajo a la izquierda; recuerde que cuando utiliza el paquete `tikz` trata al diagrama como a una imanen) y la posición de los elementos en el mismo, es probablemente más conveniente generar las imágenes en otro soporte y subirlas a `\LaTeX`.



```
\newpage
\section{Bibliografía}
\printbibliography[heading=subbibintoc,keyword={Share},title={Internet}]

\end{document}
```

%Aquí se muestra el contenido del documento en BIB, el que debe estar en el directorio de trabajo.

```
@article{einstein,
  author = "Albert Einstein",
  title = "{Zur Elektrodynamik bewegter K{\"o}rper}. ({German})
  [{On} the electrodynamics of moving bodies]",
  journal = "Annalen der Physik",
  volume = "322",
  number = "10",
  pages = "891--921",
  year = "1905",
  DOI = "http://dx.doi.org/10.1002/andp.19053221004",
  keywords = "physics"
}
```

% Este archivo muestra el formato particular para colocar referencias en un .bib, por ejemplo, un artículo

% @article{...}. Le dice que la información que se introduce es sobre un artículo. % einstein. Es el "label" asignado para llamar a esta bibliografía (citar) en el texto, la identifica de manera biunívoca.

% La información en este archivo puede luego ser impresa y referida en el documento principal con el comando \addbibresource{ejemplo.bib}

```
@book{dirac,
  title={The Principles of Quantum Mechanics},
  author={Paul Adrien Maurice Dirac},
  isbn={9780198520115},
  series={International series of monographs on physics},
  year={1981},
  publisher={Clarendon Press},
  keywords = {physics}
}
```

```
@online{knuthwebsite,
  author = "Donald Knuth",
  title = "Knuth: Computers and Typesetting",
  url = "http://www-cs-faculty.stanford.edu/~uno/abcde.html",
  keywords = "latex, knuth"
}
```

```
@inbook{knuth-fa,
  author = "Donald E. Knuth",
  title = "Fundamental Algorithms",
  publisher = "Addison-wesley",
  year = "1973",
  chapter = "1.2",
  keywords = "knuth, programming"
}
```

```
@online{sharelatexwebsite,
  author = "Share Latex",
  title = "ShareLaTeX guides",
  url = "https://es.sharelatex.com/learn",
  keywords = "Share"
}
```