

Tablas en L^AT_EX

Versión 1.00

Raúl Mata Botana
email: matabotantar (at) gmail.com

agosto del 2008

Copyright

Copyright (c) 2008 Raúl Mata Botana. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes ni Textos de Cubierta Delantera ni Textos de Cubierta Trasera. Una copia de la licencia está incluida en la sección titulada GNU Free Documentation License.

1. Introducción

Muchos tutoriales describen la construcción de tablas en L^AT_EX, pero la mayoría de ellos se limitan a explicar las opciones básicas. En este documento intentaré mostrar las distintas opciones que existen para el diseño de tablas y ejemplificando cada caso.

2. El entorno *tabbing*

Este entorno de trabajo no genera exáctamente tablas, pero permite presentar texto encolumnado, de manera similar a como lo haría un tabulador.

Del entorno de trabajo se entra y se sale mediante los comandos `\begin{tabbing}` y `\end{tabbing}` respectivamente. Cuenta con los siguientes comandos:

<code>\=</code>	Fija la posición de los tabuladores
<code>\></code>	Salta al siguiente tabulador
<code>\\</code>	Fin de línea
<code>\kill</code>	Si una línea finaliza con este comando no se imprime
<code>\+</code>	Hace que la línea sig. empiece en el tabulador sig. al previsto
<code>\-</code>	Hace que la línea sig. empiece en el tabulador anterior al previsto
<code>\<</code>	Salta al tabulador anterior
<code>\'</code>	Hace que el texto anterior se justifique a derecha
<code>\‘</code>	Hace que el texto que sigue vaya hacia el margen derecho

2.1. Ejemplo 1

Si ponemos:

```
\begin{tabbing}
Nombre \= Apellido \= Padrón\\
Esteban \> Quito \> 80000\\
Elena \> Nito \> 80001\\
Olga \> Sana \> 80002\\
\end{tabbing}
```

Lo que obtendremos es:

Nombre	Apellido	Padrón
Esteban	Quito	80000
Elena	Nito	80001
Olga	Sana	80002

2.2. Ejemplo 2

También podemos fijar el ancho de las columnas:

```
\begin{tabbing}
\hspace*{2cm} \= \hspace*{5cm} \= \hspace*{3cm} \kill
Nombre \> Apellido \> Padrón\\
Esteban \> Quito \> 80000\\
Elena \> Nito \> 80001\\
Olga \> Sana \> 80002\\
\end{tabbing}
```

Obteniendo:

Nombre	Apellido	Padrón
Esteban	Quito	80000
Elena	Nito	80001
Olga	Sana	80002

2.3. Ejemplo 3

Ahora una opción más compleja

```
\begin{tabbing}
\hspace{5cm} \= \hspace{5cm} \= \kill
Columna uno \> Columna 2 \> Columna 3 \\\
\> Segunda \> Tercera \\\
\hspace{1.5cm} \\\ %línea en blanco
Texto de primera columna que pisa la segunda \>\> Tercer columna \\\
\> Texto segunda y tercera columna \\\
\hspace{1.5cm} \\\ %línea en blanco
xxxxxxxxxxx \= xxxxxxxxxxxx \= xxxxxxxxxxxx \= \kill %seteo nuevos tabuladores
Col 1 \>Col 2 \> Col 3 \> Col 4 \+\\
Dos \> Tres \> Cuatro \\\
\< Uno \> Dos \> Tres \> Cuatro\\
Dos \> Tres \> Cuatro \- \\\
Uno \> Dos \> Tres \> Cuatro\\
Uno \> Dos \' \> Tres \' Cuatro\\
\end{tabbing}
```

Columna uno	Columna 2 Segunda	Columna 3 Tercera
Texto de primera columna que pisa la segunda	Tercer columna	
	Texto segunda y tercera columna	

Col 1	Col 2	Col 3	Col 4
	Dos	Tres	Cuatro
Uno	Dos	Tres	Cuatro
	Dos	Tres	Cuatro
Uno	Dos	Tres	Cuatro
Uno	Dos	Tres	Cuatro

3. El entorno *tabular*

Este entorno de trabajo nos permite trabajar con muchas más posibilidades que el entorno `tabbing`. Tiene dos formatos posibles que son los siguientes:

```
\begin{tabular}[posición]{columnas}
columna 1 & columna 2 & columna n \\
...
...
...
\end{tabular}
```

ó

```
\begin{tabular*}{ancho}[posición]{columnas}
columna 1 & columna 2 & columna n \\
...
...
...
\end{tabular*}
```

columnas	<p>Determina la cantidad de columnas y su posición</p> <p>l alinea a izquierda</p> <p>c centra</p> <p>r alinea a derecha</p> <p> colocado entre dos columnas genera una línea vertical</p> <p>p{ancho} Establece el ancho de una columna</p> <p>@{texto} Inserta el texto entre columnas</p> <p>@{\hspace{ancho}} Inserta espacio entre columnas</p> <p>*{num}{cols} num establece la cantidad de columnas y cols el formato de las columnas (que puede ser una expresión)</p>
posición	<p>Determina la posición de la tabla con respecto al texto</p> <p>b pone la parte inferior de la tabla en la base del texto (opción por defecto)</p> <p>c pone la mitad de la tabla en la base del texto</p> <p>t pone la parte superior de la tabla en la base del texto</p>
ancho	Establece el ancho de la tabla
&	Salta a la siguiente columna
\\	Salta a la siguiente fila
\hline	Genera una línea horizontal
\cline(n-m)	Genera una línea horizontal desde el inicio de la columna n al fin de la m
\vline	Genera una línea vertical a lo largo de una fila

Una vez que ya hemos detallado las distintas opciones del entorno tabular pasemos a ver algunos ejemplos.

3.1. Ejemplo 1

Tabla básica sin borde. Al poner:

```
\begin{tabular}{l c r}
columna 1 & columna 2 & columna 3 \\
col 1 & col 2 & col 3\\
\end{tabular}
```

Veremos entonces la primer columna alineada a izquierda, la segunda centrada y la tercera alineada a derecha:

columna 1	columna 2	columna 3
col 1	col 2	col 3

3.2. Ejemplo 2

Tabla básica con borde doble en el exterior y simple adentro. Al poner:

```
\begin{tabular}{||l | c | r||}
\hline
\hline
columna 1 & columna 2 & columna 3 \\
\hline
col 1 & col 2 & col 3\\
\hline
\end{tabular}
```

Veremos entonces:

columna 1	columna 2	columna 3
col 1	col 2	col 3

3.3. Ejemplo 3

Veamos un ejemplo un poco más complejo, crearemos una tabla de 3 filas por 3 columnas, que las columnas 1 y 2 estén separadas por “xx”, que la columna 3 esté separada en filas.

```

\begin{tabular}{l @{ xx } c @{\hspace{1cm}}|p{4cm}|}
\cline{3-3}
columna 1 & columna 2 & columna 3   esto lo pongo de relleno para
remarcar que el ancho de la columna es de 4 cm\\\cline{3-3}
col 1 & col 2 & col 3   \\\cline{3-3}
columnita 1 & columnita 2 & columnita 3   \\\cline{3-3}
\end{tabular}

```

Veremos entonces:

columna 1 xx columna 2

col 1 xx col 2

columnita 1 xx columnita 2

columna 3 esto lo pongo de relleno para remarcar que el ancho de la columna es de 4 cm
--

col 3

columnita 3

3.4. El entorno “tabular*”

Veamos un pequeño ejemplo:

`@{\extracolsep {longitud}}` Nos permite establecer un espacio en blanco entre columnas de la longitud dada.

```

\begin{tabular*}{10 cm}{|l|l@{\extracolsep{\fill}}r|}
\hline
&Desde&Hasta\\
\hline
Ruta 2&Buenos Aires & Mar del Plata\\
Ruta 7&Buenos Aires &Mendoza\\
\hline
\end{tabular*}

```

	Desde	Hasta
Ruta 2	Buenos Aires	Mar del Plata
Ruta 7	Buenos Aires	Mendoza

3.5. Alineación de decimales

Este es un ejemplo muy sencillo, la parte entera la alineamos a derecha, la decimal a izquierda y los separamos por “@{.}”.

```
\begin{tabular}{|l | r@{,}l |}  
\hline  
Producto 1 & 11 & 1\\  
\hline  
Producto 2 & 2 & 22\\  
\hline  
Total & 13 & 32\\  
\hline  
\end{tabular}
```

Producto 1	11,1
Producto 2	2,22
Total	13,32

3.6. El paquete *dcolumn* [3]

Con el método que usamos en el ejemplo anterior, si bien logramos el resultado esperado, en realidad lo que hicimos fue utilizar un pequeño truco donde uníamos 2 columnas.

Esto puede tener alguna complicación, por ejemplo si la columna que lleva una alineación decimal tiene un encabezado.

Para solucionar este problema existe el paquete *dcolumn*.

La sintaxis es la siguiente:

`D{separador 1}{separador 2}{decimales}`

separador 1 Es el que usaremos en el código para separar la parte entera de la decimal.

separador 2 Es lo que L^AT_EX nos mostrará en la salida para separar la parte entera de la decimal.

decimales Es la cantidad de decimales que se mostrarán en la salida, si el valor es “-1” no se limitará la cantidad de ciales en la salida.

Previamente, en el encabezado debemos declarar el paquete:

```
\usepackage{dcolumn}
```

Veamos como sería el ejemplo anterior:

```
\begin{tabular}{|l | D{.}{,}{-1} |}  
\hline  
Producto 1 & 11.1\\  
\hline  
Producto 2 & 2.22\\  
\hline  
Total & 13.32\\  
\hline  
\end{tabular}
```

Producto 1	11,1
Producto 2	2,22
Total	13,32

3.7. Alineación vertical de la tabla

3.7.1. Alineación de la parte superior de la tabla con la base del texto

En el siguiente ejemplo podremos ver como la tabla queda alineada con la base del texto.

```
Hola mundo  
\begin{tabular}[t]{|l |c |r|}  
\hline  
pos 1 & pos 2 & pos 3 \\  
\hline  
posición 1 & posición 2 & posición 3 \\  
\hline  
\end{tabular}
```

Hola mundo

pos 1	pos 2	pos 3
posición 1	posición 2	posición 3

3.7.2. Alineación del centro de la tabla con la base del texto

En otro ejemplo vemos como la tabla queda centrada verticalmente con el texto.

```
Hola mundo  
\begin{tabular}[c]{|l |r| c|}  
\hline
```



```
pos 1 & pos 2 & pos 3 \\
\hline
posición 1 & posición 2 & posición 3 \\
\hline
\end{tabular}
```

Hola mundo	pos 1	pos 2	pos 3
	posición 1	posición 2	posición 3

3.7.3. Alineación de la base de la tabla con la base del texto

```
Hola mundo
\begin{tabular}[b]{| 1 | c | r |}
\hline
pos 1 & pos 2 & pos 3 \\
\hline
posición 1 & posición 2 & posición 3 \\
\hline
\end{tabular}
```

Hola mundo	pos 1	pos 2	pos 3
	posición 1	posición 2	posición 3

3.8. Alineación diferenciada en una misma columna

En los ejemplos que vimos hasta ahora la alineación de una misma columna era la misma para todas sus celdas.

Si necesitamos una alineación particular para cada celda, lo que podemos hacer es poner el texto dentro de una caja.

Veamos un ejemplo:

```
\begin{tabular}{|l|l|}
\hline
celda 1 & celda 2 \\
\hline
celda 3 & \makebox[2.5cm][c]{celda 4} \\
\hline
celda 5 & \makebox[2.5cm][r]{celda 6} \\
\hline
\end{tabular}
```

Obtenemos como resultado:

celda 1	celda 2
celda 3	celda 4
celda 5	celda 6

4. Uniendo celdas

4.1. Celdas de varias columnas

Si queremos unir varias columnas, debemos usar el comando “multicolumn”. Este tiene la siguiente sintaxis:

```
\multicolumn{columnas}{posición}{texto}
```

columnas Indica cuantas columnas tendrá de ancho la celda.
posición Indica la alineación del texto.
texto Indica el contenido de la celda.

4.1.1. Ejemplo celdas multicolumnas

```
\begin{tabular}{|c|c|c|}  
\hline  
\multicolumn{3}{|c|}{multicolumna 1-3}\\  
\hline  
\multicolumn{2}{|c|}{multicolumna 1-2} & columna3\\  
\hline  
columna 1 & \multicolumn{2}{|c|}{multicolumna 2-3}\\  
\hline  
\end{tabular}
```

multicolumna 1-3		
multicolumna 1-2	columna3	
columna 1	multicolumna 2-3	

4.2. Uniendo varias filas

Para unir varias filas debemos usar el comando “multirow” [9]. Este tiene la siguiente sintaxis:

```
\multirow{filas}{ancho}[movimiento vertical]{texto}
```

Además el paquete “multirow” debe ser declarado al principio del documento.

```
\usepackage{multirow}
```

columnas	Indica cuantas filas tendrá de alto la celda.
ancho	Indica el ancho de la columna, si se pone asterisco tendrá el ancho por defecto.
movimiento vertical	Sirve para alinear el texto verticalmente(opcional).
texto	Indica el contenido de la celda.

4.2.1. Ejemplo celdas multifilas

```
\begin{tabular}{|c|c|c|}
\hline
\multirow{3}{4cm}{multifila 1-3} & \multirow{2}{*}[3 mm]
{multifila 1-2} & columna 3\\
\cline{3-3}
& & \multirow{2}{*}[-3 mm]{multifila 2-3}\\
\cline{2-2}
& columna 2 & \\
\hline
\end{tabular}
```

multifila 1-3	multifila 1-2	columna 3
	columna 2	multifila 2-3

5. Dividiendo celdas en diagonal

Para poder hacer divisiones diagonales en las tablas debemos usar el paquete “slashbox”, por lo tanto debemos agregar en el encabezado del archivo fuente la siguiente declaración:

```
\usepackage{slashbox}
```

Cuando tenemos que dividir una celda usamos el siguiente comando:

```
\backslashbox{izquierda}{derecha}
```

Donde “izquierda” y “derecha” será el contenido de la celda a cada lado de la línea diagonal.

Veamos un ejemplo:

```
\begin{tabular}{|l|r|r|r|}
\hline
\backslashbox{origen}{destino} & Buenos Aires & Córdoba & Rosario \\
\hline
Buenos Aires & 0 Km & 716 Km & 318 Km \\
\hline
Córdoba & 716 Km & 0 Km & 398 Km \\
\hline
Rosario & 318 Km & 398 Km & 0Km \\
\hline
\end{tabular}
```

<div>destino</div> <div>origen</div>	Buenos Aires	Córdoba	Rosario
Buenos Aires	0 Km	716 Km	318 Km
Córdoba	716 Km	0 Km	398 Km
Rosario	318 Km	398 Km	0Km

6. Pongamos un poco de color

Para poder colorear las tablas debemos usar el paquete *colortbl* [2], por lo tanto debemos agregar en el encabezado del archivo fuente:

```
\usepackage{colortbl}
```

6.1. Columnas en color

Para colorear las columnas disponemos del comando “columncolor”, el mismo posee la siguiente sintaxis:

```
\columncolor[color model]{color}[left overhang][right overhang]
```

color model	Especifica el modelo de color a usar, estos pueden ser rgb (red, green, blue), cmyk (cyan, magenta, yellow, black), hsb(Hue, Saturation, Brightness) o gray (gris). “hsb” no puede ser usado con pdfL ^A T _E X
color	Es una especificación al modelo de color dado. Esto es muy conveniente si uno quiere utilizarlo sin definirlo anteriormente. Los colores predefinidos son: black, white, red, green, blue, cyan, magenta y yellow.
left overhang	Controla la distancia que el color debe exceder hacia la izquierda la entrada de la tabla más ancha.
right overhang	Controla la distancia que el color debe exceder hacia la derecha la entrada de la tabla más ancha.

El comando `columncolor` solo puede ser usado en la definición de una columna o con el entorno `multicolumn`. El modo de usarlo es:

```
>\columncolor[model color]{color} ...}
```

6.1.1. Distintos modelos de color

Ahora veremos un ejemplo en el que aplicamos todos los modelos de color, en la columna 1 utilizamos el modelo rgb, en la columna 2 cmyk, en la columna 3 gray, en la columna 4 un color predefinido y en la columna 5 un color que definimos previamente.

```
\definecolor{micolor}{rgb}{0,1,0.5}

\begin{tabular}{|>\columncolor[rgb]{0.7,0,0.7}} c |
>\columncolor[cmyk]{0.8,0.5,0.4,0.1}}c |
>\columncolor[gray]{0.7}}c |
>\columncolor{blue}}c |
>\columncolor{micolor}} c|}
\hline
Col 1 & Col 2 & Col 3 & Col 4 & Col 5 \\
\hline
rgb & cmyk & gray & predefinido & definido por nosotros\\
\hline
\end{tabular}
```

Col 1	Col 2	Col 3	Col 4	Col 5
rgb	cmyk	gray	predefinido	definido por nosotros

6.1.2. Coloreando el entorno multicolumn

Veamos directamente un pequeño ejemplo:

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|>\columncolor{red}c|}{multicolumna 1-2}\\
\hline
\multicolumn{1}{|>\columncolor{green}c|}{columna 1} &
\multicolumn{1}{|>\columncolor{yellow}c|}{columna 2}\\
\hline
\end{tabular}
```

multicolumna 1-2	
columna 1	columna 2

6.1.3. Determinando el ancho del color

Como vimos en el ejemplo anterior si no definimos nada, el color ocupa todo el ancho de la columna. Ahora veremos un ejemplo que nos muestra como modificar eso.

```
\begin{tabular}{|>\columncolor[rgb]{1,1,0}1|
>\columncolor[rgb]{0,1,1}[0cm][0cm]1|
>\columncolor[rgb]{1,0,1} [.5\tabcolsep][.5\tabcolsep]1|}
\hline
Enrique & Cerse\\
\hline
Clara & Boya\\
\hline
Ana & Conda\\
\hline
\end{tabular}
```

Integrante 1	Enrique	Cerse
Integrante 2	Clara	Boya
Integrante 3	Ana	Conda

En este ejemplo vimos los dos extremos, la primer columna totalmente coloreada, la segunda columna con el color ajustado al texto y la tercer columna colorea hasta la mitad del espacio que queda entre la palabra más ancha y el ancho de la columna.

Como siempre las unidades de medida pueden ser mm, cm, in, pt, em y ex.

6.2. Filas en color

La sintaxis es similar a la de las columnas, pero más sencilla:

```
\rowcolor[model color]{color}
```

Los parámetros “model color” y “color” tienen los mismos significados que en “columncolor”.

Veamos un ejemplo:

```
\begin{tabular}{|l|l|}
\hline
\rowcolor{cmyk}{1,1,0,0}Abraham & Lapuerta\\
\hline
\rowcolor{rgb}{0,1,1}Roque & Fort\\
\hline
\rowcolor{gray}{0.9}Eva & Dirse\\
\hline
\end{tabular}
```

Abraham	Lapueta
Roque	Fort
Eva	Dirse

6.3. Coloreando celdas individuales

El paquete *colortbl* también permite colorear las celdas una a una.

La sintaxis es la siguiente:

```
\cellcolor[modelo color]{color}
```

Veamos un ejemplo sencillo:

```
\begin{tabular}{|l|l|}
\hline
\cellcolor{cmyk}{1,1,0,0}Abraham & \cellcolor{red}Lapueta\\
\hline
\cellcolor{rgb}{0,1,1}Roque & \cellcolor{blue}Fort\\
\hline
\cellcolor{gray}{0.9}Eva & \cellcolor{green}Dirse\\
\hline
\end{tabular}
```

Obtenemos como resultado:

Abraham	Lapuerta
Roque	Fort
Eva	Dirse

7. Elementos flotantes

Si nosotros generásemos una tabla con el entorno `tabular`, esta intentará ubicarse debajo del texto en el que se encuentra en el código fuente, si no entrase en la página actual se colocará en la siguiente, dejando en blanco el espacio en la hoja anterior. Esto no quedaría muy bien, pero lo podemos solucionar con el entorno “`table`”, que lo que hace es convertir la tabla en flotante. Su sintaxis es la siguiente:

```
\begin{table}[posición]
\begin{tabular}
...
...
...
\end{tabular}
\caption[Descripción corta]{Descripción larga}
\end{table}
```

`posición` Especifica donde L^AT_EX intentará colocar la tabla, tiene las siguientes opciones y se puede poner una secuencia de varias:

- b: Intentará ponerla en el fondo de la página
- h: Intentará ponerla en la misma posición en que se encuentra en el código fuente
- t: Intentará ponerla al principio de la página
- p: La pondrá en una página que contenga solo elementos flotantes
- !: Ignorar la mayoría de las restricciones impuestas por L^AT_EX

`caption` Especifica una leyenda para la tabla

Descripción larga: Leyenda que aparecerá bajo la tabla

Descripción corta: Leyenda alternativa que aparecerá en la lista de tablas

Realicemos un pequeño ejemplo:

```
\begin{table}[\hbt]
\begin{center}
\begin{tabular}{|l|l|}
...
```



```
\hline
Nombre & Apellido\\
\hline
Juan & Perez\\
José & Lopez\\
Carlos & García\\
\hline
\end{tabular}
\caption{Listado de alumnos}
\end{center}
\end{table}
```

Y obtendremos:

Nombre	Apellido
Juan	Perez
José	Lopez
Carlos	García

Tabla 1: Listado de alumnos

8. Rotando las tablas

Existe más de una manera de rotar una tabla, aquí voy a explicar las dos que yo uso, dependiendo de lo que precise. Si queremos rotar una tabla (o algunas otras cosas) debemos usar el paquete *rotating* [10], por lo tanto debemos agregar en el encabezado del archivo fuente la siguiente declaración:

```
\usepackage{rotating}
```

La primer opción es usar el entorno *sideways*, lo que hace simplemente es rotar la tabla 90°.

Para verlo mejor hagamos un ejemplo:

```
\begin{center}
\begin{sideways}
\begin{tabular}{|l|r|r|}
\hline
```

```

&columna 1&columna 2\\
\hline
fila 1 & $a_{11}$ & $a_{12}$\\
\hline
fila 2 & $a_{21}$ & $a_{22}$\\
\hline
\end{tabular}
\end{sideways}
\end{center}

```

Como resultado conseguimos lo siguiente:

	columna 1	columna 2
fila 1	a_{11}	a_{12}
fila 2	a_{21}	a_{22}

Otra manera es crear un entorno flotante análogo al entorno *table*, esto se logra mediante el entorno *sydewatstable*. Tiene una desventaja, siempre ocupará una página completa, por lo tanto si la hoja anterior no está completa la llenará con el texto que le sigue a la tabla.

Veamos un ejemplo:

```

\begin{sidewaystable}
\begin{center}
\begin{tabular}{|l|r|r|}
\hline
&columna 1&columna 2\\
\hline
fila 1 & $a_{11}$ & $a_{12}$\\
\hline
fila 2 & $a_{21}$ & $a_{22}$\\
\hline

```

```

\end{tabular}
\caption{Esta tabla está rotada.}
\label{tablaside}
\end{center}
\end{sidewaystable}

```

El resultado podemos verlo en la tabla [2](#).

8.1. Rotando el texto de las celdas

Para rotar el texto de una celda también podemos usar el entorno *sideways*. No tiene complicaciones, mostremos un ejemplo directamente.

```

\begin{center}
\begin{tabular}{|l|r|r|}
\hline
&\begin{sideways}columna 1\end{sideways}&\begin{sideways}columna 2
\end{sideways}\\
\hline
fila 1 & $a_{11}$ & $a_{12}$\\
\hline
fila 2 & $a_{21}$ & $a_{22}$\\
\hline
\end{tabular}
\end{center}

```

Como resultado obtenemos lo siguiente:

	columna 1	columna 2
fila 1	a_{11}	a_{12}
fila 2	a_{21}	a_{22}

9. Redimensionando tablas

Es probable que en alguna ocasión necesitemos cambiar el tamaño de nuestra tabla. Para esto el paquete *graphicx* nos proporciona dos comandos *scalebox* y *resizebox*.

Para utilizarlos debemos declarar en el encabezado el paquete *graphicx* [6](#).

	columna 1	columna 2
fila 1	a_{11}	a_{12}
fila 2	a_{21}	a_{22}

Tabla 2: Esta tabla está rotada.

```
\usepackage{graphicx}
```

9.1. El comando *scalebox*

Este comando permite escalar una tabla, tiene el siguiente formato:

```
\scalebox{escala horizontal}[escala vertical]{argumento}
```

Donde:

- escala horizontal: Es el factor por el que se multiplicará el ancho original de la tabla. En caso de que no se especifique el escalado vertical, este factor se aplicará tanto a la escala horizontal como vertical.
- escala vertical: Es el factor por el que se multiplicará la altura original de la tabla.
- argumento: Es lo que queremos escalar, en nuestro caso la tabla

9.1.1. Ejemplo 1

Especificando diferentes escalas para el ancho y alto de la tabla.

```
\scalebox{1.5}[2]{  
\begin{tabular}{|r|l|l|}  
\hline  
&columna 1&columna 2\\  
\hline  
fila 1& celda 1&celda 2\\  
\hline  
fila 2& celda 3&celda 4\\  
\hline  
\end{tabular}}
```

Obtenemos como resultado:

	columna 1	columna 2
fila 1	celda 1	celda 2
fila 2	celda 3	celda 4

9.1.2. Ejemplo 2

Especificando una misma escala para el ancho y alto de la tabla.

```
\scalebox{0.5}{
\begin{tabular}{|r|l|l|}
\hline
&columna 1&columna 2\\
\hline
fila 1& celda 1&celda 2\\
\hline
fila 2& celda 3&celda 4\\
\hline
\end{tabular}}
```

Obtenemos como resultado:

	columna 1	columna 2
fila 1	celda 1	celda 2
fila 2	celda 3	celda 4

9.1.3. Ejemplo 3

Si ponemos un valor negativo invertirá la tabla.

```
\scalebox{-1}{
\begin{tabular}{|r|l|l|}
\hline
&columna 1&columna 2\\
\hline
fila 1& celda 1&celda 2\\
\hline
fila 2& celda 3&celda 4\\
\hline
\end{tabular}}
```

Obtenemos como resultado:

fila 2	celda 3	celda 4
fila 1	celda 1	celda 2
columna 2	columna 1	

9.2. El comando *resizebox*

Este comando permite especificar las dimensiones reales en que quedará el objeto. Tiene 2 formatos, su sintaxis es la siguiente:

```
\resizebox{ancho}{altura}{argumento}  
\resizebox*{ancho}{altura}{argumento}
```

Según la documentación del paquete *graphicx*, el parámetro *altura* especifica la altura del objeto. Pero si el objeto fue rotado el parámetro *altura* se refiere a la altura más la profundidad. Para que el parámetro *altura* especifique solo la altura aunque la tabla esté rotada, debe usarse el comando *resizebox** en lugar de *resizebox*.

A pesar de lo que dice la documentación del paquete *graphicx*, no pude hacer que *resizebox* obtenga la altura correcta, en cambio si pude lograrlo con *resizebox**. Por lo tanto en este tutorial siempre usaré este último.

Donde:

ancho:	Define el ancho de la tabla.
altura:	Define la altura de la tabla.
argumento:	Es el objeto que queremos redimensionar, en nuestro caso la tabla.

Cuando en los parámetros *ancho* o *altura* se escribe el signo “!” en lugar de una medida indica que mantenga las proporciones respecto a la medida especificada.

9.2.1. Ejemplo 1

Acá especificamos ancho y alto de la tabla igual a 3 cm.

```
\resizebox*{3cm}{3cm}{  
\begin{tabular}{|r|l|l|}  
\hline  
&columna 1&columna 2\\  
\hline  
fila 1& celda 1&celda 2\\  
\hline  
fila 2& celda 3&celda 4\\  
\hline  
\end{tabular}}
```

Obtenemos como resultado:

	columna 1	columna 2
fila 1	celda 1	celda 2
fila 2	celda 3	celda 4

9.2.2. Ejemplo 2

En este ejemplo solo especificamos la altura y se calcula el ancho de manera proporcional.

```
\resizebox*{!}{3cm}{  
\begin{tabular}{|r|l|l|}  
\hline  
&columna 1&columna 2\\  
\hline  
fila 1& celda 1&celda 2\\  
\hline  
fila 2& celda 3&celda 4\\  
\hline  
\end{tabular}}
```

Obtenemos como resultado:

	columna 1	columna 2
fila 1	celda 1	celda 2
fila 2	celda 3	celda 4

9.2.3. Ejemplo 3

En este ejemplo especificamos un ancho igual al 80 % del ancho total del texto.

```
\resizebox*{.8\textwidth}{!}{  
\begin{tabular}{|r|l|l|}  
\hline  
&columna 1&columna 2\\  
\hline
```



```
fila 1& celda 1&celda 2\\
\hline
fila 2& celda 3&celda 4\\
\hline
\end{tabular}}
```

Obtenemos como resultado:

	columna 1	columna 2
fila 1	celda 1	celda 2
fila 2	celda 3	celda 4

10. Tablas largas

Si intentásemos crear un tabla que ocupe más de una página, veríamos que L^AT_EX la genera de manera errónea. Es por este motivo que se creó el entorno *longtable* [1]. Para poder usarlo, debemos definir en la cabecera del documento:

```
\usepackage{longtable}
```

El entorno *longtable* se puede ver como una mezcla de los entornos “tabular” y “table” (que veremos más adelante). Las columnas se definen de igual manera que en el entorno “tabular” y las filas se separan por `\\`, con la salvedad que se puede poner una unidad de longitud después del comando por ejemplo:

```
\\[3cm]
```

Esto generará un salto de línea de la medida especificada entre corchetes. No se permite alinear la tabla con respecto al texto con los argumentos [t], [b] o [c] como en el entorno tabular.

Acepta los siguientes comandos:

<code>endfirsthead</code>	Es la cabecera de la primera fila solo en la primer página.
<code>endhead</code>	Es la cabecera de la primera fila para todas las páginas salvo la primera.
<code>endlastfoot</code>	Es el pie de tabla para la última página.
<code>endfoot</code>	Es el pie de tabla para todas las páginas salvo la última.
<code>caption</code>	Numerará la tabla y le pondrá la leyenda entre llaves.
<code>caption[]</code>	Numerará la tabla, le pondrá la leyenda entre llaves y en la lista de tablas especificará el texto entre corchetes, si no se especifica entre corchetes no se agragará a la lista de tablas.
<code>caption*</code>	Pondrá la leyenda entre llaves, pero no la agragará a la lista de tablas.
<code>footnote</code>	Pone una nota al pie, no puede ser usado en cabeceras o pie de tabla.
<code>footnotemark</code>	Pone una marca de nota al pie en la cabecera o pie de tabla.
<code>footnotetext</code>	Pone la nota al pie, se debe usar en el cuerpo de la tabla.

```

\begin{longtable}{|l|l|}
\hline
\multicolumn{2}{|c|}{Primera fila de primera hoja} \\
\hline
\endfirsthead
\hline
\multicolumn{2}{|c|}{Primera fila de todas las hojas} \\
\hline
\endhead
\hline
\multicolumn{2}{|c|}{Última fila de la última hoja\footnotemark[2]}
\\
\hline
\caption{Tabla larga}
\endlastfoot
\hline
\multicolumn{2}{|c|}{Última fila de todas hojas} \\
\hline
\endfoot
\footnotetext[2]{footnote del pie de tabla.}
uno & dos\\[4cm]
\hline
tres\footnote{footnote, no puede ser usado en el entorno
‘‘tabular’’} & cuatro\\
\hline
cinco & seis\\[2cm]

```

`\end{longtable}`

Primera fila de primera hoja	
uno	dos
tres ¹	cuatro
cinco	seis
Última fila de la última hoja ²	

Tabla 3: Tabla larga

11. Tablas de ancho fijo

11.1. El paquete *tabularx* [\[4\]](#)

El paquete *tabularx* permite definir el ancho de una tabla de la misma manera que con el entorno *tabular**. Sin embargo, el entorno *tabular** tiene un problema, si el texto es más ancho que la tabla nos dará un resultado indeseado como vemos en el siguiente ejemplo:

```
\begin{tabular*}{5cm}{|l|r|}
\hline
Alumno & Padrón\\
\hline
Carlos Gardel & 80546\\
```

²footnote del pie de tabla.

¹footnote, no puede ser usado en el entorno “tabular”.

```

\hline
Aníbal Troilo & 80547\\
\hline
Homero Manzi & 80548\\
\hline
Enrique Santos Discépolo & 80549\\
\hline
\end{tabular*}

```

Alumno	Padrón
Carlos Gardel	80546
Aníbal Troilo	80547
Homero Manzi	80548
Enrique Santos Discépolo	80549

Esto se debe a que *tabular** consigue el ancho deseado completando con espacios en blanco, pero si el ancho del texto fuese mayor que el ancho de la tabla, *tabular** no puede resolver el problema.

Para remediar este inconveniente, lo que hace el paquete *tabularx* es ajustar el ancho de las columnas indicadas con la letra “X”.

Para poder usar el paquete *tabularx*, previamente debemos declararlo en el encabezado:

```
\usepackage{tabularx}
```

El entorno *tabularx* solo se aplicará a las columnas indicadas con la letra “X”.

Vemos ahora como cambia el ejemplo anterior en un entorno *tabularx*:

```

\begin{tabularx}{5cm}{|X|r|}
\hline
Alumno & Padrón\\
\hline
Carlos Gardel & 80546\\
\hline
Aníbal Troilo & 80547\\
\hline
Homero Manzi\footnote{El paquete \emph{tabularx} permite el uso de
notas al pie} & 80548\\
\hline
Enrique Santos Discépolo & 80549\\
\hline
\end{tabularx}

```

Alumno	Padrón
Carlos Gardel	80546
Aníbal Troilo	80547
Homero Manzi ²	80548
Enrique Santos Discépolo	80549

El paquete *tabularx* tiene problemas de compatibilidad con el entorno *verbatim*.

11.2. El paquete *tabulary* [5]

El paquete *tabularx*, sólo permite que el texto de las columnas cuyo ancho se ajustan automáticamente quede alineado a izquierda. Para corregir este inconveniente, existe el paquete *tabulary*.

Como siempre debemos declarar el paquete en el encabezado:

```
\usepackage{tabulary}
```

Veamos un ejemplo:

```
\begin{tabulary}{10cm}{|L|C|R|J|}
\hline
Ejemplo de celda alineada a izquierda & Ejemplo de celda con texto
centrado & Ejemplo de celda alineada a derecha & Ejemplo de celda
contexto justificado\\
\hline
\end{tabulary}
```

Ejemplo de celda alineada a izquierda	Ejemplo de celda con texto centrado	Ejemplo de celda alineada a derecha	Ejemplo de celda con texto justifi- cado
--	--	--	---

Si hay mucha diferencia de tamaño entre las columnas, el resultado que muestra el entorno *tabulary* puede no ser el esperado, ya que algunas columnas pueden ser demasiado anchas y otras demasiado angostas.

Para solucionar este problema podemos definir un ancho máximo y un ancho mínimo para las columnas.

Para definir el ancho mínimo usamos el parámetro:

²El paquete *tabularx* permite el uso de notas al pie

`\tymax=`

Por omisión este parámetro viene definido como:

`\tymax=0pt`

Para definir el ancho máximo usamos el parámetro:

`\tymax=`

Por omisión este parámetro viene definido como:

`\tymax=\maxdimen`

11.2.1. Algunos inconvenientes con *tabulary*

Este paquete parece tener problemas cuando el ancho del texto no es mayor que el ancho de la hoja, de manera que no respeta el ancho que se define.

Veamos por ejemplo el mismo ejemplo que usamos en *tabularx*:

```
\begin{tabulary}{5cm}{|L|R|}  
\hline  
Alumno & Padrón\\  
\hline  
Carlos Gardel & 80546\\  
\hline  
Aníbal Troilo & 80547\\  
\hline  
Homero Manzi & 80548\\  
\hline  
Enrique Santos Discépolo & 80549\\  
\hline  
\end{tabulary}
```

Alumno	Padrón
Carlos Gardel	80546
Aníbal Troilo	80547
Homero Manzi	80548
Enrique Santos Discépolo	80549

Podemos apreciar que el ancho real es 6,79 cm, y no 5 cm como habíamos indicado.

Otro inconveniente que tiene este paquete es que las notas al pie no funcionan muy bien.

El entorno *multicolumn* puede ser usado, si el texto no supera el ancho calculado de la columna.

12.1. El paquete *booktabs*[\[7\]](#)

Para poder usar este paquete previamente debemos especificarlo en el emcabecera:

Incorpora 5 nuevos comandos, que son los siguientes:

<code>\toprule[espesor]:</code>	se utiliza para líneas cabecera, opcionalmente se puede especificar un espesor determinado.
<code>\midrule[espesor]:</code>	se utiliza para líneas que separan filas comunes, opcionalmente se puede especificar un espesor determinado.
<code>\bottomrule[espesor]:</code>	se utiliza para líneas de fin de tabla, opcionalmente se puede especificar un espesor determinado.
<code>\cmidrule[espesor]</code> (recorte) <code>{cola-colb}</code> :	se utiliza para crear líneas que no atraviesen todas las columnas, opcionalmente se puede especificar un espesor determinado. El parámetro <i>recorte</i> especifica si la línea se recorta a la izquierda y/o a la derecha y en que proporción. Los parámetros <i>cola</i> y <i>colb</i> indican de que columna a que columna va la línea.
<code>\addlinespace[espesor]:</code>	se usa para especificar espacio en blanco entre filas, opcionalmente se puede especificar un espesor determinado.

Veamos un ejemplo simple:

página 31 de 37

```

\midrule
fila 1& celda 11 & celda 12\\
\cmidrule{2-3}
fila 2& celda 21 & celda 22\\
\addlinespace
fila 3 & celda 31 &celda 32\\
\bottomrule
\end{tabular}

```

Obtenemos como resultado:

	columna 1	columna 2
fila 1	celda 11	celda 12
fila 2	celda 21	celda 22
fila 3	celda 31	celda 32

12.1.2. Ejemplo 2

En el ejemplo anterior vimos un ejemplo básico con los parámetros por omisión. Cambiemos ahora alguno de esos valores, de manera bien exagerada para que la diferencia sea evidente.

```

\begin{tabular}{l c r}
\toprule[3mm]
&columna 1 &columna 2\\
\midrule[2mm]
fila 1& celda 11 & celda 12\\
\cmidrule[4mm]{2-3}(lr{5mm})
fila 2& celda 21 & celda 22\\
\cmidrule{2-3}
fila 3& celda 31 & celda 32\\
\addlinespace[10mm]
fila 4 & celda 41 &celda 42\\
\bottomrule[5mm]
\end{tabular}

```

Obtenemos como resultado:

	columna 1	columna 2
fila 1	celda 11	celda 12
fila 2	celda 21	celda 22
fila 3	celda 31	celda 32
fila 4	celda 41	celda 42

12.1.3. Ejemplo 3

Un ejemplo más real podría ser el siguiente:

```
\begin{tabular}{l r r r}
\toprule[0.8mm]
&enero&febrero &total\\
\cmidrule(1){2-4}
Juan&40& 60 & 100 \$\\
\midrule
Pedro&70& 30 & 100 \$\\
\cmidrule[0.5mm](1){4-4}
&&&200 \$ \\
\bottomrule[0.8mm]
\end{tabular}
```

Obtenemos como resultado:

	enero	febrero	total
Juan	40	60	100 \$
Pedro	70	30	100 \$
			200 \$

13. Cambiando el aspecto de las líneas

Existen una serie de comandos que nos permiten cambiar el aspecto de las líneas. Para definirlos debemos usar el siguiente formato:

```
\setlength{\parametro}{valor}
\begin{tabular}
....
\end{tabular}
```

Descripción de los comandos:

- arrayrulewidth: Es el grosor de las líneas de la tabla. Por omisión este valor es *0,4pt*.
- doublerulesep: Es la separación entre líneas dobles, ya sean estas horizontales o verticales.
- tabcolsep: Es la mitad del ancho de separación entre dos columnas.

Si quisieramos cambiar la distancia entre filas debemos usar el comando *arraystretch*:

- arraystretch: Es la separación entre filas. Su valor por omisión es 1. Si pusieramos por ejemplo 1.5, aumentaríamos la separación un 50.

Lo definimos de la siguiente manera:

```
\renewcommand{\arraystretch}{valor}
```

Estas modificaciones pueden hacerse en el preámbulo, donde afectarán a todo el documento o dentro del entorno, donde solo afectará a este.

Veamos algunos ejemplos:

13.1. Ejemplo 1

```
%Acá cambiamos la separación entre columnas
{\setlength{\tabcolsep}{10mm}
%Acá cambiamos el grosor de las líneas
{\setlength{\arrayrulewidth}{2mm}
\begin{tabular}{| 1 | 1 |}
\hline
celda 1 & celda 2\\
\hline
celda 3 & celda 4\\
\hline
\end{tabular}
}}
```

celda 1	celda 2
celda 3	celda 4

13.2. Ejemplo 2

```
\begin{center}
%Acá cambiamos la separación entre filas
\renewcommand{\arraystretch}{2.5}
%Acá cambiamos la separación entre filas dobles
{\setlength{\doublerulesep}{2mm}
\begin{tabular}{|| 1 | 1 ||}
\hline
\hline
celda 1 & celda 2\\
\hline
celda 3 & celda 4\\
\hline
\hline
\end{tabular}
}
\end{center}
```

celda 1	celda 2
celda 3	celda 4

14. Cambiando la leyenda de las tablas

Si en el entorno “table” usamos el comando “caption”, este pondrá la leyenda “Cuadro” con el número de tabla correspondiente. En castellano es más común llamarlo “Tabla” en lugar de “Cuadro”, esto podemos modificarlo con el siguiente comando después de `\begin{document}`:

```
\renewcommand{\tablename}{Tabla}
```

Si además queremos cambiar “Índice de cuadros” por “Índice de tablas”, debemos usar el comando:

```
\renewcommand{\listtablename}{Índice de tablas}
```

15. Resultados inesperados

Este inconveniente me lo hizo notar *Leg Babouyes*.

Supongamos que queremos colorear una celda multifila como en el siguiente ejemplo:

```
\begin{tabular}{|>{\columncolor{red}}c|c|}
\hline
\multirow{2}{2cm}{celda} & celda 2\\
\cline{2-2}
& celda 3\\
\hline
\end{tabular}
```

El resultado que obtendremos, será:

celda	celda 2
	celda 3

Esto se debe a que las celdas se colorean una a una, entonces, la segunda celda de la multifila se colorea después de haber escrito el texto que se hizo junto con la primer celda, por eso queda tapado.

Una de las posibles soluciones pasa por escribir el texto en la segunda celda y desplazarlo hacia arriba, de la siguiente manera:

```
\begin{tabular}{|>{\columncolor{red}}c|c|}
\hline
& celda 2\\
\cline{2-2}
\multirow{2}{2cm}[5mm]{celda} & celda 3\\
\hline
\end{tabular}
```

celda	celda 2
	celda 3

Referencias

- [1] David Carlisle. *The longtable package*, noviembre 1996.
- [2] David Carlisle. *The colortbl package*, mayo 1998.

- [3] David Carlisle. *The dcolumn package*, septiembre 1999.
- [4] David Carlisle. *The tabularx package*, enero 1999.
- [5] David Carlisle. *The tabulary package*, septiembre 2003.
- [6] David Carlisle. *Packages in the graphics bundle*, noviembre 2005.
- [7] Simon Fear. *Publication quality tables in L^AT_EX*, abril 2005.
- [8] Indian TEX Users Group. *L^AT_EX Tutorials, A PRIMER*, septiembre 2003.
- [9] Jerry Leichter and Piet van Oostrum. *multirow.sty - Span multiple rows of a table*.
- [10] Herbert Vob. *Rotating Text, Tabulars and Images*, marzo 2007.