

Apuntes de Latex

Capitulo 3: Más conceptos básicos: Listas, tablas, figuras

1. Listas

1.1. Entorno itemize

El entorno `itemize` permite escribir una lista de objetos, siendo adecuado para listas sencillas. Las diversas entradas de la lista se resaltan con un indicador, que por defecto es un círculo negro (`\textbullet`), aunque el uso de la opción `spanish` de `babel` puede cambiar esta predefinición.

Obsérvese el siguiente ejemplo:

```
\begin{itemize}
  \item Esto es un ejemplo de una lista
  \item Las distintas entradas se introducen con el comando
  \verb+\item+
  \item Si una entrada es muy larga, obsérvese cómo el párrafo
    continua manteniendo una indentación fija determinada por la
    posición del indicador.
\end{itemize}
```

que produce:

- Esto es un ejemplo de una lista.
- Las distintas entradas se introducen con el comando `\item`.
- Si una entrada es muy larga, obsérvese cómo el párrafo continua manteniendo una indentación fija determinada por la posición del indicador.

Las listas pueden anidarse, con entradas que posean subentradas, hasta una profundidad de cuatro subniveles (más anidamiento no está contemplado, y obtendríamos un error de compilación). Por ejemplo:

```
\begin{itemize}
\item Primera entrada del primer nivel
\item Segunda entrada del primer nivel
  \begin{itemize}
    \item Primera entrada del segundo nivel
    \item Segunda entrada del segundo nivel
  \end{itemize}
  \vdots
\end{itemize}
```

⋮

```

\begin{itemize}
\item Primera entrada del tercer nivel
\item Segunda entrada del tercer nivel
  \begin{itemize}
    \item Primera entrada del cuarto nivel
    \item Segunda entrada del cuarto nivel
  \end{itemize}
\end{itemize}
\end{itemize}

```

- Primera entrada del primer nivel
- Segunda entrada del primer nivel
 - Primera entrada del segundo nivel
 - Segunda entrada del segundo nivel
 - * Primera entrada del tercer nivel
 - * Segunda entrada del tercer nivel
 - Primera entrada del cuarto nivel
 - Segunda entrada del cuarto nivel

Como indicábamos al principio, el uso de `\usepackage[spanish]{babel}` cambia los indicadores predefinidos para cada nivel (•, –, *, · para L^AT_EX estándar) a: ■, ●, ○, ◇. L^AT_EX nos permite cambiar a nuestro gusto estos marcadores, cosa que podemos hacer con la siguientes instrucciones:

- `\renewcommand{\labelitemi}{Nuevo-marcador}`
- `\renewcommand{\labelitemii}{Nuevo-marcador}`
- `\renewcommand{\labelitemiii}{Nuevo-marcador}`
- `\renewcommand{\labelitemiv}{Nuevo-marcador}`

donde Nuevo-marcador designa al código L^AT_EX del nuevo indicador, y `\labelitemi`, `\labelitemii`, etc... representan respectivamente los diferentes niveles de anidamiento. Por ejemplo, incluyendo el paquete `pifont` de símbolos especiales podemos conseguir:

```

\renewcommand{\labelitemi}{\ding{42}}
\renewcommand{\labelitemii}{\ding{43}}
\begin{itemize}
\item Primera entrada del primer nivel
\item Segunda entrada del primer nivel
  \begin{itemize}
    \item Primera entrada del segundo nivel
    \item Segunda entrada del segundo nivel
  \end{itemize}
\end{itemize}

```

- ☛ Primera entrada del primer nivel
- ☛ Segunda entrada del primer nivel
 - ☛ Primera entrada del segundo nivel
 - ☛ Segunda entrada del segundo nivel

Nótese que si colocamos el comando `\renewcommand{\labelitemi}{Nuevo-marcador}` *fuera* de un entorno `itemize`, afectará a todas las posteriores listas `itemize` del documento. Si solamente se desea que afecte a una lista en particular, las nuevas redefiniciones se deben colocar *dentro* de entorno, de la forma siguiente:

```
\begin{itemize}
\renewcommand{\labelitemi}{\ding{42}}
\item Primera entrada del primer nivel
\item Segunda entrada del primer nivel
\begin{itemize}
\renewcommand{\labelitemii}{\ding{43}}
\item Primera entrada del segundo nivel
\item Segunda entrada del segundo nivel
\end{itemize}
\end{itemize}
```

Finalmente, podemos modificar la distancia entre entradas de la lista con el comando: `\itemsep=distancia`, que suma `distancia` a la separación predefinida. Si `distancia` es negativa, reducirá la distancia entre entradas. Éste comando debe introducirse inmediatamente *después* de `\begin{itemize}`:

```
\begin{itemize}\itemsep=-6pt
\item Primera entrada del primer nivel
\item Segunda entrada del primer nivel
\item Tercera entrada del primer nivel
\begin{itemize}
\item Primera entrada del segundo nivel
\item Segunda entrada del segundo nivel
\end{itemize}
\end{itemize}
```

- Primera entrada del primer nivel
- Segunda entrada del primer nivel
- Tercera entrada del primer nivel
- Primera entrada del segundo nivel
- Segunda entrada del segundo nivel

1.2. Entorno enumerate

Para obtener listas ordenadas, se utiliza el entorno `enumerate`, que igualmente es anidable hasta cuatro subniveles:

```
\begin{enumerate}
\item Primera entrada del primer nivel
\item Segunda entrada del primer nivel
\begin{enumerate}
\item Primera entrada del segundo nivel
\item Segunda entrada del segundo nivel
\begin{enumerate}
\item Primera entrada del tercer nivel
\item Segunda entrada del tercer nivel
\begin{enumerate}
\item Primera entrada del cuarto nivel
\item Segunda entrada del cuarto nivel
\end{enumerate}
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

1. Primera entrada del primer nivel
2. Segunda entrada del primer nivel
 - a) Primera entrada del segundo nivel
 - b) Segunda entrada del segundo nivel
 - 1) Primera entrada del tercer nivel
 - 2) Segunda entrada del tercer nivel
 - a' Primera entrada del cuarto nivel
 - b' Segunda entrada del cuarto nivel

Advirtamos que las etiquetas para cada tipo de nivel son cambiadas por la opción `spanish` de `babel`. El ejemplo anterior ilustra el resultado en tales circunstancias. Para \LaTeX estándar se obtendría:

- 1., 2., 3., ...
- (a), (b), (c), ...
- i., ii., iii., ...
- A., B., C., ...

Existen métodos para manipular las etiquetas, pero para describirlos necesitamos conocimientos relativamente avanzados, por lo que los abordaremos más adelante. Sin embargo, existe una alternativa simple y elegante, usando el paquete `enumerate`. Éste paquete introduce un argumento optativo en el entorno `enumerate`, de la forma: `\begin{enumerate}[argumento-optativo]`, donde en el argumento se debe elegir un carácter clave: `1`, `i`, `I`, `a` y `A`, (números, números romanos, y letras) sobre el cual se basará la enumeración. Lo elegante del paquete es que permite combinar el carácter clave con instrucciones \LaTeX . Veamos un ejemplo:

```
Veamos ahora los pasos necesarios
para escribir un documento en \LaTeX:
\begin{enumerate}[\hspace*{0.5cm}]%
\bfseries P{a}so 1]
  \item Preparar documento fuente tex
  \item Compilarlo para producir dvi
  \begin{enumerate}[(a)]
    \item Visualizar con un visor de dvi
    \item Corregir errores
    \item Recompilar
  \end{enumerate}
  \item Convertir a PostScript con dvips
\end{enumerate}
```

Veamos ahora los pasos necesarios para escribir un documento en \LaTeX :

Paso 1 Preparar documento fuente

Paso 2 Compilarlo para producir dvi

- (a) Visualizar con un visor de dvi
- (b) Corregir errores
- (c) Recompilar

Paso 3 Convertir a PostScript con dvips

¿Para qué se ha utilizado `\hspace*{0.5cm}` en el argumento optativo? La razón reside en que, a diferencia del entorno `itemize`, en el entorno `enumerate` se suprime la sangría para el primer nivel de elementos. Otro punto importante es el porqué de la ‘a’ entre llaves en “Paso”. Es interesante ver qué ocurre si eliminamos las llaves (ejercicio).

1.3. Entorno `description`

El entorno `description` se puede considerar como una generalización del entorno `itemize`, en la que las etiquetas pueden ser libremente configuradas, lo cual es adecuado para descripciones. Por ejemplo:

```
\begin{description}
  \item[\TeX] Un procesador de textos
  \item[Word] Otro procesador de textos
  \item .....
\end{description}
```

\TeX Un procesador de textos

Word Otro procesador de textos

.....

Se observa que, por defecto, las etiquetas son escritas en negrita. Podemos cambiar esto sin problemas, por ejemplo:

<code>\begin{description}</code>	<code>\TeX</code> Un procesador de textos
<code>\renewcommand{\descriptionlabel}[1]{</code>	
<code>{\hspace*{0.5cm}\textsf{#1}}</code>	
<code>\item[\TeX] Un procesador de textos</code>	<code>Word</code> Otro procesador de textos
<code>\item[Word] Otro procesador de textos</code>	
<code>\item</code>
<code>\end{description}</code>	

lo cual pone los objetos a describir en **san serif**, además de introducir una pequeña sangría (ausente en el primer nivel, al igual que en **enumerate**).

Es importante mencionar que en cualquiera de los otros entornos (**itemize** y **enumerate**) puede en cualquier momento cambiarse la etiqueta por defecto, añadiendo la nueva etiqueta entre paréntesis tras el comando `\item`:

<code>\begin{itemize}</code>	■ Primera entrada del primer nivel
<code>\item Primera entrada del primer nivel</code>	
<code>\item[\rightarrow] Segunda entrada del primer nivel</code>	→ Segunda entrada del primer nivel
<code>\begin{itemize}</code>	
<code>\item Primera entrada del segundo nivel</code>	
<code>\item[\clubsuit] Segunda entrada del segundo nivel</code>	● Primera entrada del segundo nivel
<code>\end{itemize}</code>	♣ Segunda entrada del segundo nivel
<code>\end{itemize}</code>	

Finalmente, destaquemos que todos estos tipos de listas pueden combinarse anidándose listas de distintos tipos:

<code>\begin{enumerate}</code>	1. Puede mezclar los entornos de listas a su gusto:
<code>\item Puede mezclar los entornos de listas a su gusto:</code>	
<code>\begin{itemize}</code>	
<code>\item Pero podría comenzar a parecer incómodo.</code>	■ Pero podría comenzar a parecer incómodo.
<code>\item Si abusa de ellas.</code>	■ Si abusa de ellas.
<code>\end{itemize}</code>	
<code>\item Por lo tanto, recuerde:</code>	2. Por lo tanto, recuerde:
<code>\begin{description}</code>	
<code>\item[Lo innecesario] no va a resultar adecuado porque lo coloque en una lista.</code>	Lo innecesario no va a resultar adecuado porque lo coloque en una lista.
<code>\item[Lo adecuado,] sin embargo, se puede presentar agradablemente en una lista.</code>	Lo adecuado , sin embargo, se puede presentar agradablemente en una lista.
<code>\end{description}</code>	
<code>\end{enumerate}</code>	

2. Tablas

Para la construcción de tablas, la herramienta básica es el entorno `tabular`. La estructura fundamental de una tabla es la siguiente:

```
\begin{tabular}[Posición]{FormatoColumnas}
xxx & xxx & xxx & xxx & ... & xxx \\
xxx & xxx & xxx & xxx & ... & xxx \\
... & ... & ... & ... & ... & ... \\
xxx & xxx & xxx & xxx & xxx & xxx
\end{tabular}
```

& Separador entre columnas, que marca el fin de una casilla y el principio de la siguiente. Podemos dejar una casilla vacía con `&&`. Nótese que el tamaño de las columnas no tiene nada que ver con la distancia entre los separadores y el texto de la casilla: `&xxx&`, `& xxx &` ó `& xxx &` producirán exactamente el mismo resultado. En general, la anchura de la columna está determinada por longitudes predefinidas y, fundamentalmente, por el tamaño máximo que alcanzan los elementos de una columna.

**** Cambio de fila: es la instrucción para comenzar una fila nueva. No es estrictamente necesaria para la última fila, a menos que se quiera terminar con una línea horizontal (comando: `\\ \hline`)

Posición Argumento **optativo** que especifica la posición de la tabla respecto al texto en el que se incluye. Puede tomar los valores `t` (top), `c` (center; valor por defecto) ó `b` (bottom), según qué parte de la tabla se alinee con respecto a la línea de texto principal. Por ejemplo:

<pre>aqui se inserta \begin{tabular}[t]{cc} 11 & 12 \\ 21 & 22 \end{tabular} una pequeña tabla</pre>	<pre>aqui se inserta \begin{tabular}[c]{cc} 11 & 12 \\ 21 & 22 \end{tabular} una pequeña tabla</pre>	<pre>aqui se inserta \begin{tabular}[b]{cc} 11 & 12 \\ 21 & 22 \end{tabular} una pequeña tabla</pre>
<pre> aqui se inserta 11 12 una pe- 21 22 queña tabla</pre>	<pre> aqui se inserta 11 12 una pe- queña tabla 21 22</pre>	<pre> aqui se inserta 11 12 una pe- 21 22 queña tabla</pre>

FormatoColumnas Argumento fundamental que utilizamos para dar estructura a la tabla. Se compone de dos elementos fundamentales:

Especificadores Debe haber un especificador por cada columna de la tabla. Tenemos las siguientes opciones:

l, r, c Introduce una nueva columna justificada a la izquierda, derecha ó centro, respectivamente. La anchura de columna se determina automáticamente a partir del tamaño máximo de sus elementos.

p{Ancho} Se utiliza en caso de que tengamos un texto largo en una de las columnas, creando una columna de anchura fija *Ancho*

Separadores Son optativos, y se encargan de especificar cómo deben separarse las columnas

| Produce una barra vertical separando columnas.

@{Objeto} Suprime el espacio entre columnas e inserta en su lugar el *Objeto* declarado.

Para añadir líneas horizontales se pueden utilizar:

`\hline` Traza una línea horizontal a lo largo de toda la tabla

`\cline{x-y}` Traza una línea horizontal desde la columna x hasta la columna y, ambas inclusive

Finalmente, el comando `\multicolumn{NumeroColumnas}{FormatoColumnas}{Objeto}` se puede utilizar para agrupar, dentro de una fila, el contenido de varias columnas (argumento `NumeroColumnas`) en una sola (véase el ejemplo a continuación). Otros comandos útiles son `\extracolsep{Longitud}`, que añade un espacio adicional `Longitud` a la distancia entre *columnas* calculada automáticamente por \LaTeX , y `\setlength{\extrarowheight}{Longitud}`, que añade el espacio adicional `Longitud` a la distancia estándar entre *filas*.

2.1. Ejemplos

Importaciones (en millones de \euro) de carne y verduras:

```
\begin{tabular}{ccc}
  Pais & Carne & Verduras \\ \hline
  España & 1390 & 980 \\
  Francia & 1504 & 3020 \\
  Italia & 2010 & 1040 \\
\end{tabular}
```

Importaciones (en millones de \euro) de carne y verduras:

Pais	Carne	Verduras
España	1390	980
Francia	1504	3020
Italia	2010	1040

Imaginemos que queremos:

Planeta	Distancia al sol (millones km)	
	Máxima	Mínima
Mercurio	69.4	46.8
Venus	109.0	107.6
Tierra	152.6	147.4

Podemos intentar:

```
\begin{tabular}{|l|r|r|}
\hline
  & \multicolumn{2}{c}{Distancia al sol} \\
Planeta & \multicolumn{2}{c}{(millones km)} \\
  & Máxima & Mínima \\ \hline
Mercurio & 69.4 & 46.8 \\
Venus & 109.0 & 107.6 \\
Tierra & 152.6 & 147.4 \\ \hline
\end{tabular}
```

lo que nos daría:

Planeta	Distancia al sol (millones km)	
	Máxima	Mínima
Mercurio	69.4	46.8
Venus	109.0	107.6
Tierra	152.6	147.4

pero no queda bien!

la solución está en añadir una barra separadora | tras el argumento “c” en `\multicolumn`, con lo que quedaría:

Planeta	Distancia al sol (millones km)	
	Máxima	Mínima
Mercurio	69.4	46.8
Venus	109.0	107.6
Tierra	152.6	147.4

Ejercicios:

1) Construir la siguiente tabla, de forma que «h» y «m» se escriban automáticamente:

Salida	6 h	00 m
Control 1	6 h	45 m
Control 2	7 h	15 m

2) Contruir la siguiente tabla (pista: hace falta usar `\multicolumn`):

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Hombres	324	123	250	210
Mujeres	143	243	286	222

3) Contruir la siguiente tabla:

CONTENIDO DEL CURSO		
Capítulo	Páginas	Resumen
1. Números	1 – 8	Se establece el lenguaje básico y se definen los «conjuntos numéricos» y sus propiedades básicas.
2. Continuidad	9 – 20	Se define el concepto de función continua y se estudian la propiedades de las funciones continuas. En particular, se estudian el teorema de los valores intermedios y el teorema de Weierstrass.

4) Contruir la siguiente tabla:

Grupo	Medida (en % de respuestas correctas)							
	Edad		Lectura		Sílabas		Fonemas	
	(meses)				segmentación		segmentación	
	M	SD	M	SD	M	SD	M	SD
Experimental	86.9	3.7	82.7	10.2	87.2	10.4	81.6	13.0
Control	89.2	3.1	77.9	16.9	86.5	10.0	82.4	12.4

2.2. El entorno table

Dado que las tablas son grandes, y es complicado ubicarlas de forma elegante, tenemos a nuestra disposición el entorno `table` para manejarlas como *objetos flotantes* (aquellos para los cuales L^AT_EX busca la mejor colocación). El formato a utilizar sería:

```
\begin{table}[Posición] % Se puede elegir entre h,t,b,p para posición
\centering % (si queremos la tabla centrada con respecto al texto)
\begin{tabular}{cccc}
.....
\end{tabular}
\caption{Pie de tabla}
\label{Etiqueta para referenciar la tabla en el documento}
\end{table}
```

La forma de referenciar la tabla sería entonces añadir en cualquier lugar del documento `\ref{Etiqueta}`. Todas las tablas incluidas mediante este procedimiento son numeradas automáticamente, y es posible contruir un índice de tablas incluyendo el comando: `\listoftables`. Podemos introducir un pie de tabla mediante el comando `\caption{Pie de Tabla}`. Las opciones de posición h,t,p,b tienen el siguiente significado:

- h → Se prefiere la colocación en el lugar del documento fuente donde introducimos la tabla.
- t → Parte superior de la página
- b → Parte inferior de la página
- p → Nueva página formada exclusivamente por elementos flotantes

Veamos un pequeño ejemplo:

```
\begin{table}[ht]
\centering
\begin{tabular}{|c|c|c|} \hline
Pais & Carne & Verduras \\ \hline
España & 1390 & 980 \\ \hline
Francia & 1504 & 3020 \\ \hline
Italia & 2010 & 1040 \\ \hline
\end{tabular}
\caption{Importaciones (en millones de \euro) de carne y verduras}
\label{tabla_importaciones}
\end{table}
```

Produciría: (en éste lugar, en el documento fuente, se han incluido las líneas anteriores)

Entonces, si queremos referenciar a la tabla en cualquier parte del texto, se utilizaría: `\ref{tabla_importaciones}`, por ejemplo:

En la Tabla 1 se muestran las importaciones...

Pais	Carne	Verduras
España	1390	980
Francia	1504	3020
Italia	2010	1040

Cuadro 1: Importaciones (en millones de €) de carne y verduras

Podemos observar que, en éste ejemplo en particular, a pesar de utilizar como primera opción “h”, se utiliza la segunda opción “t”, es decir, la tabla es desplazada a la parte de arriba de la ´pagina siguiente. Ésto se debe a que en el lugar donde tentativamente queríamos incluirla no se disponía de espacio suficiente para ello.

Otro hecho curioso es la etiqueta que se coloca en el pie de tabla (Cuadro 1:). Podemos cambiar esta predefinición de la opción `spanish` de `babel`, mediante la instrucción:

`\addto\captionsspanish{\renewcommand*{\tablename}{Tabla}}` situada en el preámbulo (si queremos un efecto global) y, por supuesto, **detrás** de la declaración `\usepackage[babel]{spanish}`. Si queremos un efecto local, debemos utilizar `\renewcommand*{\tablename}{Tabla}` situado en el interior de un grupo cualquiera en el cuerpo del documento.

Finalmente, mencionemos (más información se añadirá en el capítulo correspondiente) que tenemos a nuestra disposición otro entorno completamente análogo, el entorno `figure`, para incluir y organizar figuras dentro de un documento. Las únicas diferencias son el uso de una sintaxis diferente para abrir y cerrar el entorno (`\begin{figure} ... \end{figure}`) y para incluir un índice de figuras (`\listoffigures`).

2.3. Respuestas a los ejercicios:

Ejercicio 1:

```
\begin{tabular}{|l|c@{ h \ \ }c@{ m }|}
\hline
Salida & 6 & 00 \\\
Control 1 & 6 & 45 \\\
Control 2 & 7 & 15 \\\ \hline
\end{tabular}
```

Ejercicio 2:

```
\begin{tabular}{r|c|c|c|c|}
\cline{2-5}
& Grupo 1 & Grupo 2 & Grupo 3 & Grupo 4\\\ \hline
\multicolumn{1}{|r|}{Hombres}& 324&123&250&210\\\
\hline \multicolumn{1}{|r|}{Mujeres}&143&243&286&222\\\ \hline
\end{tabular}
```

Ejercicio 3:

```
\begin{tabular}{lr@{\,--\,}lp{5.8cm}}
\multicolumn{4}{c}{\bf CONTENIDO DEL CURSO}\hline\hline
\multicolumn{1}{c}{\bf Capítulo}
& \multicolumn{2}{c}{\bf Páginas}
& \multicolumn{1}{l}{\bf Resumen}\hline\hline
\textbf{1. Números} & 1 & 8 & %
Se establece el lenguaje básico y se definen los <<conjuntos
numéricos>> y
sus propiedades básicas.\hline
\textbf{2. Continuidad} & 9 & 20 & %
Se define el concepto de función continua y se estudian la
propiedades de las funciones continuas.
\newline
En particular, se estudian el teorema de los valores intermedios y
el teorema de Weierstrass.\hline
\end{tabular}
```

Ejercicio 4:

```
\begin{tabular}{lrrrrrrrrrrrr}
& \multicolumn{11}{c}{Medida (en \% de respuestas correctas)}\hline
& \multicolumn{2}{c}{Edad} & & \multicolumn{2}{c}{Lectura}& & \multicolumn{2}{c}{Sílabas} & & \multicolumn{2}{c}{Fonemas} & \\
& \multicolumn{2}{c}{(meses)}& & \multicolumn{2}{c}{ } & & \multicolumn{2}{c}{segmentación} & & \multicolumn{2}{c}{segmentación} & \\
\cline{2-3} & \cline{5-6} & \cline{8-9} & \cline{11-12}
Grupo & M & SD & & M & SD & & M & SD & & M & SD & \\
\hline
Experimental & 86.9 & 3.7 & & 82.7 & 10.2 & & 87.2 & 10.4 & & 81.6 & 13.0 & \\
Control & 89.2 & 3.1 & & 77.9 & 16.9 & & 86.5 & 10.0 & & 82.4 & 12.4 & \\
\hline
\end{tabular}
```

2.4. Herramientas y paquetes útiles para tablas

Parámetros de control de formato de tablas

Podemos modificar la apariencia de una tabla con los parámetros descritos a continuación. Nótese que, si queremos hacer el efecto de un cambio de parámetros *local*, deberíamos incluir la instrucción de la siguiente forma:

```
{
\setlength{\NombreParámetro}{ValorNuevo}
\begin{tabular}
```

```
....
\end{tabular}
}
```

esto es, incluyéndola dentro de un grupo que agrupe el entorno tabular.

- `\arraystretch` → Factor que controla la separación vertical entre filas; el valor por defecto es 1, igual al valor de `\baselinestretch`. Definiéndolo como 1.5 obtendremos una tabla con columnas un 50 % más altas. Debido a que `\arraystretch` es un *comando*, debemos redefinirlo con la sintaxis: `\renewcommand*{\arraystretch}{NuevoFactor}`
- `\tabcolsep` → 1/2 de la separación horizontal entre columnas, así como el espacio horizontal al comienzo y final de la tabla. Dado que es una *longitud*, se redefine mediante: `\setlength{\tabcolsep}{NuevaSeparación}`
- `\arraycolsep` → Igual que el anterior, para el entorno matemático `array`. Siendo longitud, se redefine análogamente al caso anterior. Para el resto, también longitudes, se aplica lo mismo.
- `\arrayrulewidth` → Grosor de las líneas horizontales y verticales en el entorno `tabular`. Por defecto, 0.4 pt.
- `\doublerulesep` → Separación entre rayas dobles, horizontales ó verticales.

Unos ejemplos para ilustrar lo anterior:

```
\begin{center}
{\setlength{\tabcolsep}{10pt}
\begin{tabular}{||c|c||}\hline
e & ef \\
efg & efgh \\ \hline
\end{tabular}}
\par \bigskip
{\setlength{\arrayrulewidth}{3pt}
\renewcommand*{\arraystretch}{2}
\begin{tabular}{|c|c|}\hline
i & ij \\ \hline
ijk & ijkl \\ \hline
\end{tabular}} \par \bigskip
{\setlength{\arrayrulewidth}{2pt}
\setlength{\doublerulesep}{2pt}
\begin{tabular}{||c|c||}\hline
m & mn \\
mno & mnop \\ \hline
\end{tabular}}
\end{center}
```

e	ef
efg	efgh

i	ij
ijk	ijkl

m	mn
mno	mnop

Debemos destacar que, si no se utiliza el paquete `array`, (que se describirá a continuación) el uso de líneas gruesas crea problemas obteniéndose uniones imperfectas.

2.5. El paquete array

El paquete array introduce una serie de mejoras sobre el entorno `tabular` estándar. La tabla siguiente describe los nuevos elementos introducidos por el paquete:

Argumentos de los entornos estándar <code>tabular</code> y <code>array</code>	
<code>l</code> , <code>r</code> , <code>c</code> , <code>p</code> {Ancho}	Formatos de columna, que mantienen su significado básico. Para <code>p</code> {Ancho}, el texto se coloca justificado a la parte de arriba de la casilla.
<code> </code> <code>@</code> {Objeto}	Mantienen su significado, aunque se introduce una mejora para <code> </code> : el espacio entre columnas se incrementa en el valor correspondiente a la anchura de la raya vertical.
Nuevos argumentos con el paquete array	
<code>m</code> {Ancho}	Nuevo especificador de columna, similar a <code>p</code> {Ancho}, con la diferencia de que las entradas aparecen centradas en sentido vertical.
<code>b</code> {Ancho}	Similar al anterior, pero las entradas se justifican a la parte inferior.
<code>!</code> {Objeto}	Nuevo separador de columnas, similar a <code>@</code> {Objeto}. La diferencia está en que no suprime el espacio normal entre columnas.
<code>></code> {comando}	Se debe incluir antes de un especificador <code>l</code> , <code>r</code> , <code>c</code> , <code>p</code> , <code>m</code> , <code>b</code> . Tiene el efecto de aplicar el comando a cada una de las casillas de la columna.
<code><</code> {comando}	Análogo al anterior, sólo que se utiliza <i>después</i> del especificador de columna, y aplica el comando detrás de cada columna.

Para casillas tipo párrafo (`p`{Ancho}, `m`{Ancho} y `b`{Ancho}), es posible dividir el texto en varias líneas, y justificarlo a cualquiera de los lados utilizando los comandos:

- `\raggedright` → El texto se ajusta sólo por la izquierda
- `\centering` → Texto centrado
- `\raggedleft` → El texto se ajusta sólo por la derecha

Dentro de cada casilla, cada línea se separa utilizando el comando estándar `\\`. Es importante tener en cuenta que, para la última casilla de una fila, se debe terminar con el comando `\tabularnewline`, que indica a \LaTeX que hemos finalizado con esa fila de la tabla.

Ejemplo:

```
\begin{tabular}{|l|c|} \hline
\multicolumn{1}{|p{1.5cm}|}{\centering %
Planeta} & \multicolumn{1}{p{4cm}|}{%
\centering Distancia media al sol \\
(millones de km)} \tabularnewline \hline
Mercurio & 58.1 \\
Venus & 108.3 \\
Tierra & 150.0 \\ \hline
\end{tabular}
```

Planeta	Distancia media al sol (millones de km)
Mercurio	58.1
Venus	108.3
Tierra	150.0

Cambiando los formatos “p{1.5cm}” y “p{4cm}” por “m{1.5cm}” y “m{4cm}” ó por “b{1.5cm}” y “b{4cm}” obtendríamos, respectivamente:

Planeta	Distancia media al sol (millones de km)
Mercurio	58.1
Venus	108.3
Tierra	150.0

Planeta	Distancia media al sol (millones de km)
Mercurio	58.1
Venus	108.3
Tierra	150.0

Vemos en los ejemplo anteriores que los puntos de los decimales no quedan bien alineados. Una posible solución sería colocar el signo “.” como separador de columna, de la siguiente forma:

```
\begin{tabular}{|l|r@{.}l|}
\hline
\multicolumn{1}{|m{1.5cm}|}{\centering %
Planeta} & \multicolumn{2}{m{4cm}|}{%
\centering Distancia media al sol \\
(millones de km)} \tabularnewline \hline
Mercurio & 58&1 \\
Venus & 108&3 \\
Tierra & 150&1 \\
\hline
\end{tabular}
```

Planeta	Distancia media al sol (millones de km)
Mercurio	58.1
Venus	108.3
Tierra	150.1

Sin embargo, tal procedimiento no da buenos resultados para columnas muy anchas. La solución, para este caso, se encuentra en utilizar el paquete dcolumn. Este paquete introduce un nuevo separador, de sintaxis:

`D{separador1}{separador2}{Num}`

donde **separador1** es el decimal que utilizamos en el fichero .tex, y **separador2** es el decimal que L^AT_EX colocará en la salida. Num denota el número máximo de decimales en la columna (el valor -1 equivaldría a cualquiera). Por ejemplo, obtendríamos el resultado deseado con:

```
\begin{center}
\begin{tabular}{|l|D{,}{.}{-1}|}
\hline
\multicolumn{1}{|m{1.5cm}|}{\centering %
Planeta} & \multicolumn{1}{m{4cm}|}{%
\centering Distancia media al sol \\
(millones de km)} \tabularnewline \hline
Mercurio & 58,1 \\
Venus & 108,3 \\
Tierra & 150,1 \\
\hline
\end{tabular}
\end{center}
```

Planeta	Distancia media al sol (millones de km)
Mercurio	58.1
Venus	108.3
Tierra	150.1

Los argumentos `>{comando}` y `<{comando}` nos permiten hacer útiles manipulaciones por columnas.

Por ejemplo, si queremos poner en negrita, añadir color, etc... una columna en concreto, podemos especificar:

```
\begin{center}
\begin{tabular}{|>\bfseries l|}%
>\color{red} c|}
\hline
\multicolumn{1}{|m{1.5cm}|}{\centering %
Planeta} & \multicolumn{1}{m{4cm}|}%
{\centering Distancia media al sol \\
(millones de km)} \tabularnewline \hline
Mercurio & 58.1 \\
Venus & 108.3 \\
Tierra & 150.1 \\
\hline
\end{tabular}
\end{center}
```

Planeta	Distancia media al sol (millones de km)
Mercurio	58.1
Venus	108.3
Tierra	150.1

La utilidad del comando de cierre `<\comando` puede verse en el siguiente ejemplo:

```
\begin{tabular}{|c|>{\displaystyle}c<{\}}|}
\hline
Función & \text{Definición} \\ \hline
tangente & \frac{\sen(x)}{\cos(x)} \\ \hline
cosecante & \frac{\cos(x)}{\sen(x)} \\ \hline
\end{tabular}
```

Función	Definición
tangente	$\frac{\sen(x)}{\cos(x)}$
cosecante	$\frac{\cos(x)}{\sen(x)}$

donde los contenidos de la segunda columna se escriben automáticamente en modo matemático tipo párrafo.

Si un determinado formato de columna se va a repetir en varias tablas (ó columnas), es conveniente utilizar el siguiente comando, proporcionado por el paquete array, que permite construir nuevos formatos:

```
\newcolumntype{Carácter}{Definición}
```

Así por ejemplo, la expresión `>\displaystyle c<\}` en la tabla anterior puede ser abreviada a 'X', utilizando previamente:

```
\newcolumntype{X}{>\displaystyle c<\}}
```

En tablas con líneas horizontales, las letras mayúsculas pueden quedar demasiado cerca de las líneas; para solventar ésto el paquete **array** proporciona la longitud `\extrarowheight`, que permite añadir una pequeña altura a cada línea mejorando el resultado, como puede verse en el siguiente ejemplo (donde además, se muestra la diferencia entre los separadores @ y !)

```

\newcolumnntype{L}{>{\itshape}r}
\begin{tabular}{|l|L@{---}l|}
\hline
    & Nombre & Name \\\hline
1 & uno & one \\\
2 & dos & two \\\
3 & tres & three \\\hline
\end{tabular}
\par \bigskip
\setlength{\extrarowheight}{2pt}
\begin{tabular}{|l|L!{---}l|}
\hline
    & Nombre & Name \\\hline
1 & uno & one \\\
2 & dos & two \\\
3 & tres & three \\\hline
\end{tabular}

```

	<i>Nombre</i> —Name
1	<i>uno</i> —one
2	<i>dos</i> —two
3	<i>tres</i> —three

	<i>Nombre</i> — Name
1	<i>uno</i> — one
2	<i>dos</i> — two
3	<i>tres</i> — three

2.6. El paquete multirow

El paquete `multirow` nos permite construir tablas en las que algunas celdas ocupan varias filas dentro de un entorno tabular. Se utiliza la orden:

```
\multirow{nrow}{width}[vmove]{contenido}
```

donde: `nrow` es el número de filas a agrupar; `width` es el ancho de la columna; y `vmove` sirve para subir o bajar el texto (opcional). Esta orden funciona de forma similar a `\multicolumn`, pero para filas. Nótese que el formato `m{Ancho}` nos era de utilidad para centrar texto en el caso de columnas vecinas de diferente altura; `multirow` es adecuado para centrar texto con respecto a *varias* columnas vecinas. Por ejemplo:

```

\begin{tabular}{|l|r|r|}
\hline \multirow{4}{1.8cm}{Planeta}
    & \multicolumn{2}{p{3.8cm}|}%
    {\centering Distancia al sol \\\
    (millones de km)}\tabularnewline \cline{2-3}
    & \multicolumn{1}{p{1.7cm}|}%
    {\centering Maxima \\\ distancia}
    & \multicolumn{1}{p{1.7cm}|}%
    {\centering Minima \\\ distancia}
    \tabularnewline \hline
Mercurio & 69.4 & 46.8\\
Venus & 109.0 & 107.6\\
Tierra & 152.6 & 147.4\\
\hline
\end{tabular}

```

Planeta	Distancia al sol (millones de km)	
	Maxima distancia	Minima distancia
Mercurio	69.4	46.8
Venus	109.0	107.6
Tierra	152.6	147.4

Se aprecia que la entrada “Planeta” está justificada a la izquierda. Esto se debe a que el *comando* `\multirowsetup`, que determina el modo en el que aparece el texto de `\multirow`, está definido por defecto como `\raggedright`. Redefiniéndolo con `\renewcommand*{\multirowsetup}{\centering}` se consigue un texto centrado:


```

\renewcommand{\multirowsetup}{\centering}
\begin{tabular}{|l|r|r|}
\hline \multirow{4}{1.8cm}{Planeta}
& \multicolumn{2}{p{3.8cm}|}%
{\centering Distancia al sol \\\
(millones de km)}\tabularnewline \cline{2-3}
& \multicolumn{2}{p{1.7cm}|}%
{\centering Maxima \\\ distancia}
& \multicolumn{2}{p{1.7cm}|}%
{\centering Minima \\\ distancia}
\tabularnewline \hline
Mercurio & 69.4 & 46.8\\
Venus & 109.0 & 107.6\\
Tierra & 152.6 & 147.4\\
\hline
\end{tabular}

```










Planeta	Distancia al sol (millones de km)	
	Maxima distancia	Minima distancia
Mercurio	69.4	46.8
Venus	109.0	107.6
Tierra	152.6	147.4

2.7. El paquete `hhline`

El paquete `hhline` define el comando `\hhline`, que produce rayas dobles ó simples, y añade capacidades para producir intersecciones bien construidas entre líneas horizontales y verticales. El comando se utiliza como:

```
\hhline{ColumnasEIntersecciones}
```

donde la especificación `ColumnasEIntersecciones` se hace utilizando los siguientes elementos:

-  Una raya horizontal doble del ancho de una columna.
-  Una raya horizontal simple del ancho de una columna.
-  Una columna sin raya horizontal.
-  Una raya vertical que corta a una horizontal (simple ó doble).
-  Una raya vertical que es partida por una horizontal doble. Detrás ó delante debe haber `=`.
-  Dos rayas verticales que cortan a una horizontal doble.
-  La semiparte superior de una horizontal doble.
-  La semiparte inferior de una horizontal doble.
-  Podemos usar la abreviatura: `*{3}{==#}` (por ejemplo), que se expande como `==#==#==#` (ésto también es válido para el entorno `tabular` básico).

Los elementos anteriores se utilizan agrupados, para definir entrelazamientos con las rayas verticales. Por ejemplo:

- `|t:` \longrightarrow Esquina superior izquierda del cruce de dos rayas dobles (\ulcorner)
- `:t|` \longrightarrow Como el anterior, pero para la esquina superior derecha (\urcorner)

- `|b:` → Análogo, para esquina inferior izquierda (\lrcorner)
- `:b|` → Análogo, para esquina inferior derecha (\llcorner)
- `|:` → Una raya vertical seguida de otra vertical que se cruza con una horizontal doble
- `:|` → Una raya vertical, que se ha cruzado con una horizontal doble, seguida de otra vertical
- `::` → Enlace sin cortes entre dos rayas dobles
- `||` → Raya vertical doble que no es atravesada por las rayas horizontales
- `#` → Corte de rayas dobles

El ejemplo siguiente ilustra las diferentes posibilidades de intersección:

```
\begin{tabular}{||cc||c|c||}
\hline{t:==:t:==:t|}
a&b&c&d\\
\hline{|:==:|~|~||}
1&2&3&4\\
\hline{#==#~|=#}
i&j&k&l\\
\hline{||--||--||}
w&x&y&z\\
\hline{|b:==:b:==:b|}
\end{tabular}
```

a	b	c	d
1	2	3	4
i	j	k	l
w	x	y	z

2.8. Tablas con color: el paquete `colortbl`

El objetivo del paquete `colortbl` es dar color al fondo de las tablas y a las rayas de separación. Este paquete basa su funcionamiento en los paquetes `color` y `array`, que se cargarán automáticamente al cargar `colortbl` (podemos, por tanto, incluir en `colortbl` las opciones de controlador propias de `color`).

Para dar color a filas y columnas, podemos utilizar los comandos:

- `\columncolor[Modelo]{Color}[SepIzq][SepDer]` (columnas)
- `\rowcolor[Modelo]{Color}[SepIzq][SepDer]` (filas)

Debemos hacer aquí un pequeño inciso para explicar la sintaxis extendida del comando `\color`. Podemos utilizar simplemente el comando `\color{NombreColor}`, que permite utilizar hasta 68 colores predefinidos (con la opción `usenames` del paquete `color`), ó contruir nuestros propios colores a medida mediante el comando `\color[Modelo]{Especificación-de-color}`. Podemos elegir, para `Modelo`, entre las opciones:

- `rgb` → Especificación-de-color debe ser una terna de números entre 0 y 1, cada uno de los cuales especifica la cantidad relativa de rojo, verde y azul en la mezcla
- `cmyk` → Análogo, pero utilizando 4 números que representan valores de cyan, magenta, amarillo y negro

- `gray` \longrightarrow Un valor entre 0 y 1, especificando la proximidad al blanco ó negro del tono de gris

Todas éstas posibilidades son extensibles a los comandos `\columncolor` y `\rowcolor`, por tanto. Los argumentos (optativos) `[SepIzq]` y `[SepDer]` indican las longitudes (a izquierda y derecha, respectivamente) que el fondo de color debe exceder del espacio ocupado por el texto de la celda de la tabla. Por defecto, su valor es `\tabcolsep`, lo cual quiere decir que si no se incluyen, el fondo de color ocupará toda la celda.

Cada una de las dos versiones, para fila ó columna, debe utilizarse de forma diferente:

- `\columncolor` se debe colocar como argumento del especificador de columna `>\{...\}` del paquete `array` ó en la definición de un nuevo tipo de columna a través de `\newcolumntype`. También podemos incluirlo dentro de un comando `\multicolumn` (para rellenar una sola celda, por ejemplo), pero siempre utilizando `>\{...\}` en el argumento de `\multicolumn`.
- `\rowcolor` debe aparecer SIEMPRE al comienzo de un fila. Debe tenerse en cuenta de que si entrecruzamos declaraciones de color en filas y columnas, la declaración `\rowcolor` prevalece, ya que es la última en ser asignada.

Finalmente, el color de las líneas de separación se controla mediante los comandos:

- `\arrayrulecolor[Modelo]{color}` que fija el color de las líneas
- `\doublerulesepcolor[Modelo]{color}` que fija el color del relleno entre líneas dobles

ADVERTENCIA FINAL: Sólo está asegurada la obtención del resultado correcto en el fichero final .pdf (los visores .dvi pueden dar resultados algo extraños).

Ejemplos:

```
\begin{tabular}{|>\columncolor[gray]{0.9}}1%
>\color{white}\columncolor[gray]{0.6}}r|}
alfa & beta \ \ gamma & delta
\end{tabular} \hspace{1cm}
\begin{tabular}%
{|>\color{yellow}%
\columncolor[gray]{0.7}[0pt]}1%
>\columncolor{yellow}[0.5\tabcolsep]}r|}
alfa & beta \ \ gamma & delta
\end{tabular}
```

alfa	beta	alfa	beta
gamma	delta	gamma	delta

```
\begin{tabular}{||c||c||}
\hhline{|t::t::t|}
\rowcolor{red} alfa & beta \ \
\hhline{|:>:::|}
\rowcolor{green} gamma & delta \ \
\hhline{|b::b::b|}
\end{tabular}
```

alfa	beta
gamma	delta