

Apuntes de Latex

Capítulo 2: Listas y Tablas

SECCIÓN 1

Listas no numeradas: El 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. Las entradas de la lista también se encuentran indentadas respecto al margen izquierdo, y se añade por defecto un espacio vertical antes y después de la lista. Obsérvese en el siguiente ejemplo la sintaxis del entorno y el uso del comando `\item` para introducir las entradas de la lista:

```
... párrafo anterior

\begin{itemize}
  \item Esto es un ejemplo de una lista
  \item En el documento fuente colocamos
una pequeña indentación en el comando
item, para entender más fácilmente el
código, aunque esto no es imprescindible.
  \item Si una entrada es muy larga, nótese
cómo el párrafo continua manteniendo una
indentación fija.
\end{itemize}

Párrafo posterior ...
```

... párrafo anterior

- Esto es un ejemplo de una lista
- En el documento fuente colocamos una pequeña indentación en el comando `item`, para entender más fácilmente el código, aunque esto no es imprescindible.
- Si una entrada es muy larga, nótese cómo el párrafo continua manteniendo una indentación fija.

Párrafo posterior ...

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). Para cada subnivel se utiliza un símbolo de marcación diferente, y se añade una indentación extra con respecto al nivel anterior. Observar el siguiente ejemplo donde se anidan hasta 4 subniveles:

```

\begin{itemize}
\item 1ª entrada del primer nivel
\item 2ª entrada del primer nivel
  \begin{itemize}
\item 1ª entrada del segundo nivel
\item 2ª entrada del segundo nivel
  \begin{itemize}
\item 1ª entrada del tercer nivel
\item 2ª entrada del tercer nivel
  \begin{itemize}
\item 1ª entrada del cuarto nivel
\item 2ª entrada del cuarto nivel
  \end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}

```

- 1ª entrada del primer nivel
- 2ª entrada del primer nivel
 - 1ª entrada del segundo nivel
 - 2ª entrada del segundo nivel
 - 1ª entrada del tercer nivel
 - 2ª entrada del tercer nivel
 - ◇ 1ª entrada del cuarto nivel
 - ◇ 2ª entrada del cuarto nivel

Como indicábamos al principio, el uso de `\usepackage[spanish]{babel}` cambia los indicadores predefinidos para cada nivel (\bullet , $-$, $*$, \cdot para \LaTeX estándar) a: \blacksquare , \bullet , \circ , \diamond . \LaTeX 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 \LaTeX del nuevo indicador, y `\labelitemi`, `\labelitemii`, etc... son los comandos \LaTeX que se encargan de escribir los marcadores para los primeros, segundos, etc... niveles de enumeración. Por ejemplo, incluyendo el paquete `pifont` de símbolos especiales (colocar `\usepackage{pifont}` en el preámbulo) podemos obtener lo siguiente:

```

\renewcommand{\labelitemi}{\ding{42}}
\renewcommand{\labelitemii}{\ding{43}}
\begin{itemize}
\item 1ª entrada del primer nivel
\item 2ª entrada del primer nivel
  \begin{itemize}
\item 1ª entrada del segundo nivel
\item 2ª entrada del segundo nivel
  \end{itemize}
\end{itemize}
\end{itemize}

```

- ☛ 1ª entrada del primer nivel
- ☛ 2ª entrada del primer nivel
 - ☛ 1ª entrada del segundo nivel
 - ☛ 2ª 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 pueden 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}

```

(Un método alternativo sería incluir los comandos `\renewcommand{\labelitemi}{...}` y el entorno `itemize` correspondiente dentro de un grupo, añadiendo llaves al principio y el final)

 SECCIÓN 2

Listas numeradas: El entorno enumerate

Para obtener listas numeradas, se utiliza (de forma análoga al `itemize`) el entorno `enumerate`, que igualmente es anidable hasta cuatro subniveles:

```

\begin{enumerate}
\item 1ª entrada del primer nivel
\item 2ª entrada del primer nivel
  \begin{enumerate}
  \item 1ª entrada del segundo nivel
  \item 2ª entrada del segundo nivel
  \begin{enumerate}
  \item 1ª entrada del tercer nivel
  \item 2ª entrada del tercer nivel
  \begin{enumerate}
  \item 1ª entrada del cuarto nivel
  \item 2ª entrada del cuarto nivel
  \end{enumerate}
  \end{enumerate}
  \end{enumerate}
\end{enumerate}

```

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

En éste caso, cada comando `\item` tiene el efecto de ir añadiendo entradas contabilizadas según valores crecientes de un contador. Éste contador, dependiendo del nivel de anidamiento del entorno `enumerate`, posee diversas *representaciones* (ó formatos); para el primer nivel se utiliza 1., 2., 3., etc..., para el segundo a), b), c), etc..., y otras representaciones distintas para los niveles tercero y cuarto. Es importante tener en cuenta que el formato de 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 libremente tal formato de las etiquetas de enumeración, pero para describirlos necesitamos conocimientos relativamente avanzados sobre las características de los contadores en \LaTeX , que serán abordados más adelante. De momento, pondremos una alternativa simple y elegante (aunque no muy potente), usando el paquete `enumerate` (**para lo cual, como siempre, se debe de declarar `\usepackage{enumerate}` en el preámbulo del documento.**)

Éste paquete permite colocar 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 visor de dvi
\item Corregir errores
\item Recompilar
\end{enumerate}
\item Convertir a .ps con dvips
\end{enumerate}
```

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

Paso 1 Preparar documento fuente tex

Paso 2 Compilarlo para producir dvi

(a) Visualizar con visor de dvi

(b) Corregir errores

(c) Recompilar

Paso 3 Convertir a .ps 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).

SECCIÓN 3

El 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:

```
\renewcommand{\descriptionlabel}[1] %
{\hspace*{0.5cm}\textsf{#1}}
\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

.....

lo cual pone los objetos a describir en *sanserif*, además de introducir una pequeña sangría¹ (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`:

```
\begin{itemize}
  \item 1ª entrada
  \item[ $\rightarrow$ ] 2ª entrada
  \item 3ª entrada
  \begin{enumerate}
    \item 1º entrada
    \item 2ª entrada
    \item[ $\clubsuit$ ] 3ª entrada
  \end{enumerate}
\end{itemize}
```

■ 1ª entrada

→ 2ª entrada

■ 3ª entrada

1. 1º entrada

2. 2ª entrada

♣ 3ª entrada

Como también se puede ver en el ejemplo anterior, no existe ningún problema en anidar listas de distintos tipos (respetando siempre, por supuesto, el límite de 4 niveles de anidamiento).

SECCIÓN 4

Listas personalizadas: el entorno list

A la hora de formatear las entradas de una lista, L^AT_EX utiliza ciertos valores predeterminados para la colocación de los párrafos que conforman cada entrada, la distancia de las etiquetas a la entrada, etc... Todos éstos valores (ó variables de longitud) son pueden ser modificados mediante reasignaciones de longitud con cualquiera de las sintaxis alternativas:

¹El comando `\renewcommand{\descriptionlabel}[1]{Acciones del comando}` es un ejemplo de definición de comando (en éste caso, redefinición) dependiente de un argumento; el número 1 entre corchetes indica que el comando es dependiente de 1 argumento variable, que dentro del conjunto de comandos L^AT_EX en “Acciones del comando” se denota con el símbolo clave #1. En el presente caso vemos que las acciones son dejar un espacio de 0.5cm y poner en tipo *sanserif* “#1”, ésto es, el argumento del comando `\descriptionlabel`. Se ampliarán éstos conceptos en el Capítulo dedicado a la programación en L^AT_EX.

- `\setlength{\Longitud}{Xmm}` (ó cm, pt, etc...)
- `\Longitud=Xmm`
- `\Longitud Xmm`

que asignan el nuevo valor `Xmm` a la variable `\Longitud`. Otro método de asignación de longitudes es el comando `\addtolength{\Longitud}{Xmm}` que *incrementa* en `Xmm` el valor de la variable de longitud. Éste método es particularmente útil en casos en que no conozcamos a priori el valor preestablecido de una longitud, y queramos modificarla sin arriesgarnos a introducir valores desproporcionados.

Para crear listas tipo `itemize` con parámetros configurables, podemos emplear el entorno `list`, con la siguiente sintaxis:

```
\begin{list}{Etiqueta}{Declaraciones}
\item Texto...
\item Texto...
\end{list}
```

donde el parámetro `Etiqueta` especifica el objeto que debe emplearse como etiqueta, y dentro del apartado `Declaraciones` debemos incluir comandos de redefinición de longitud. La Figura 1 muestra gráficamente, en el caso de un entorno `list` típico, las distintas longitudes que utiliza \LaTeX para estructurar la lista, y que podemos modificar a voluntad. Las longitudes `\leftmargin` y `\rightmargin` definen los “márgenes” de la lista de ítems con respecto a la anchura de texto estándar del documento. Es fundamental tener en cuenta que `\leftmargin` se define como el espacio horizontal de indentación de la lista *respecto al entorno anterior*. Su valor depende del nivel de lista en el que nos encontremos; para el primer nivel, toma el valor de la longitud `\leftmargini`, para el segundo, `\leftmarginii`, y así hasta `\leftmarginiv` para el cuarto nivel. Reasignando éstas longitudes (antes de comenzar la anidación de las listas) podemos por tanto cambiar la indentación según nuestras preferencias particulares (ver ejemplos a continuación).

Por otra parte, `\topsep` permite ajustar los espacios verticales anteriores y posteriores a la lista, `\itemsep` la separación vertical entre ítems. Para ajustar la posición de las etiquetas respecto al texto de la entrada, se puede ajustar el parámetro `\labelsep`, y para la indentación de los párrafos que conforman una entrada se utiliza `\itemindent`. Los siguientes ejemplos ilustran varias posibilidades de uso de éstos parámetros:

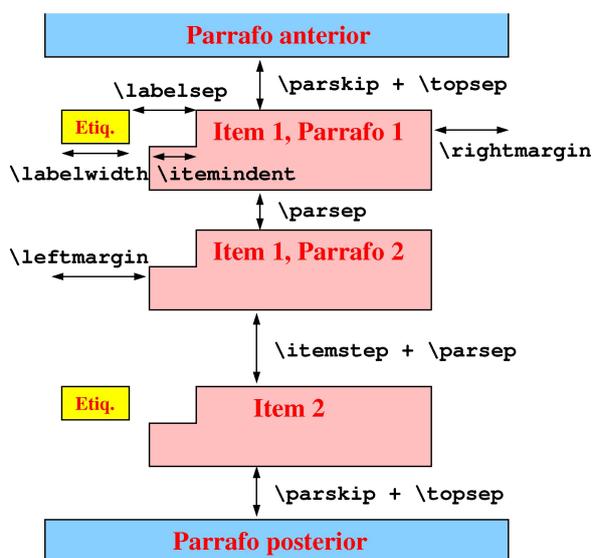


Figura 1: Variables de longitud en entornos de tipo lista

Lista estándar; nótese cómo al dejar vacío la opción para etiquetas éstas desaparecen.

```
... texto anterior
\begin{list}{}{}
  \item Primer ítem de una lista con
  valores estándar para las longitudes
  \par Segundo párrafo del primer
  ítem de la lista
  \item Segundo ítem de la lista.
  \begin{list}{}{}
    \item Primer ítem de una sublista
    anidada dentro de la lista principal
    \item Segundo ítem de la sublista
  \end{list}
\end{list}
Texto posterior ...
```

```
... texto anterior

  Primer ítem de una lista con valores estándar
  para las longitudes

  Segundo párrafo del primer ítem de la lista

  Segundo ítem de la lista.

      Primer ítem de una sublista anidada
      dentro de la lista principal

      Segundo ítem de la sublista

Texto posterior ...
```

Ejemplo donde se modifica la separación con el texto circundante (`\topsep`), la indentación de los ítems (`\itemindent`), la separación entre ítems (`\itemsep`), y la separación etiqueta-ítem (`\labelsep`).

```
... texto anterior
\begin{list}{\textbullet}{}%
\addtolength{\topsep}{3mm}%
\addtolength{\labelsep}{1mm}%
\addtolength{\itemsep}{-2mm}%
\setlength{\itemindent}{3mm}
  \item Primer ítem de una lista con
  valores de formato modificados
  \par Segundo párrafo del primer
  ítem de la lista
  \item Segundo ítem de la lista.
  \begin{list}{\scriptscriptstyle%
  \blacksquare}{}
    \item Primer ítem de una sublista
    anidada dentro de la lista principal
    \item Segundo ítem de la sublista
  \end{list}
\end{list}
Texto posterior ...
```

```
... texto anterior

  • Primer ítem de una lista con valores de
  formato modificados

  Segundo párrafo del primer ítem de la lista

  • Segundo ítem de la lista.

      ■ Primer ítem de una sublista anidada
      dentro de la lista principal

      ■ Segundo ítem de la sublista

Texto posterior ...
```

En el ejemplo siguiente, obsérvese cómo se ajustan las indentaciones para listas de primer y segundo nivel; además, se ajustan las separaciones entre ítems a través de `\itemstep`

```

\setlength{\leftmargini}{0pt}
\setlength{\leftmarginii}{30pt}
\begin{list}{\textbullet}{%
\addtolength{\itemsep}{-2mm}%
\setlength{\itemindent}{2mm}}
\item Primer ítem de una lista con
valores de formato modificados \par
Segundo párrafo del primer ítem
\item Segundo ítem de la lista.
\begin{list}{${\scriptscriptstyle}%
\blacksquare$}{\setlength{\itemindent}{%
3mm}\addtolength{\itemsep}{1mm}}
\item Primer ítem de una sublista
anidada dentro de la lista principal
\item Segundo ítem de la sublista
\end{list}
\end{list}

```

- Primer ítem de una lista con valores de formato modificados

Segundo párrafo del primer ítem

- Segundo ítem de la lista.
 - Primer ítem de una sublista anidada dentro de la lista principal
 - Segundo ítem de la sublista

Finalmente, hay que tener en cuenta que algunos de éstos parámetros de configuración no son exclusivos del entorno `list` y que pueden emplearse en los entornos `itemize` ó `enumerate`, como por ejemplo `\itemstep` ó `\leftmargini`, `\leftmarginii`, etc... Véase el siguiente ejemplo en el que se ajustan las indentaciones y el espaciado entre ítems:

```

\setlength{\leftmargini}{10mm}
\setlength{\leftmarginii}{15mm}
\begin{enumerate}
\addtolength{\itemsep}{-2mm}
\item Funciones de variable compleja
\item Teorema de Cauchy-Goursat
\begin{itemize}
\addtolength{\itemsep}{-1mm}
\item Enunciado
\item Demostración
\end{itemize}
\end{enumerate}

```

1. Funciones de variable compleja
2. Teorema de Cauchy-Goursat

- Enunciado
- Demostración

SECCIÓN 5

Construcción de tablas: el entorno tabular

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 elemento de una columna* (con excepción de las comlumnas tipo párrafo ó pancho en las que la anchura se fija de antemano).

**** 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- 21 22 queña tabla</pre> | <pre> 11 12 aqui se inserta 21 22 una pe- 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 L^AT_EX, y `\setlength{\extrarowheight}{Longitud}`, que añade el espacio adicional `Longitud` a la distancia estándar entre *filas* (se debe cargar el paquete `array` para que éste último comando funcione).

5.1. 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)} \\ \hline
  & 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 |

En éste último ejemplo se observa una de las utilidades principales del comando `\multicolumn`; aparte de servir para agrupar columnas, podemos emplearlo para cambiar el formato estándar de columna (tanto alineación como separadores) de una casilla de la tabla en particular. Pueden encontrarse más ejemplos de esto en el documento ejemplo asociado a ésta sección.

5.2. 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 \\
\end{tabular}}
\par \bigskip
{\setlength{\arrayrulewidth}{3pt}
\renewcommand*\arraystretch{2}
\begin{tabular}{|c|c|} \hline
i & ij \\
ijk & ijkl \\
\end{tabular}} \par \bigskip
{\setlength{\arrayrulewidth}{2pt}
\setlength{\doublerulesep}{2pt}
\begin{tabular}{||cc||} \hline
m & mn \\
mno & mnop \\
\end{tabular}}
\end{center}

```

| | |
|-----|------|
| e | ef |
| efg | efgh |

| | |
|-----|------|
| i | ij |
| ijk | ijkl |

| | |
|-----|------|
| m | mn |
| mno | mnop |

Debemos destacar que, si no se carga en el preámbulo el paquete `array`, (que se describirá en un capítulo posterior) el uso de líneas gruesas crea problemas obteniéndose uniones imperfectas.

5.3. La script `Excel2Latex`

Si trabajamos frecuentemente con hojas de cálculo como MS-Excel, puede sernos muy útil el instalar bajo Excel la herramienta **Excel2Latex**, que proporciona una forma sencilla de convertir nuestros datos en formato Excel a entornos `tabular` de \LaTeX . Para instalarla se debe descargar del CTAN el archivo **Excel2LaTeX.xla**

(en la dirección <http://www.ctan.org/tex-archive/support/excel2latex/>)

y posteriormente copiarlo al directorio de ADDINS de office (que normalmente se encuentra en: `C:\Documents and Settings\usuario\Datos de programa\Microsoft\AddIns`, aunque su ubicación puede variar según la ubicación de Office y la versión de Windows). Se abre entonces una vez desde Excel (habilitando la opción de utilizar macros)² una vez, y a partir de entonces lo tendremos ya disponible como una opción más en “Herramientas”.

Para su uso, simplemente se selecciona una zona de un documento excel, se ejecuta la script “Excel2LaTeX”, y obtendremos una ventana emergente con el código \LaTeX (en forma de entorno `tabular`) que podremos entonces copiar y pegar a nuestro documento \LaTeX .

²Dependiendo de la instalación de Office, puede ser necesario el bajar el nivel de seguridad para las macros en las opciones de Office