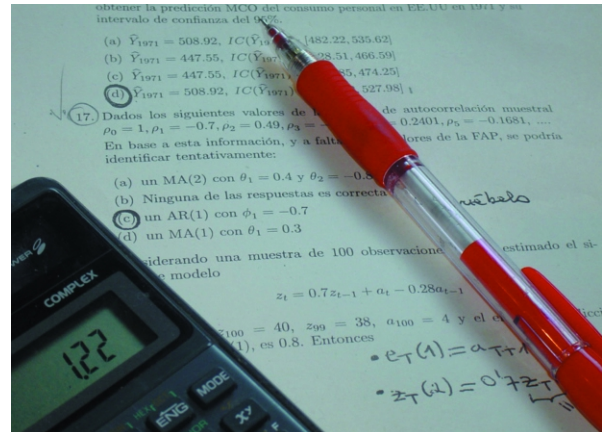


## LaTeX en Linux (3ª Parte)

# La Tabla de Multiplicar (y otras)

Con este artículo continuamos nuestro paseo por el mundo de la escritura con LaTeX. En las dos entregas anteriores hemos aprendido que LaTeX es un sistema de escritura profesional en el que, a pesar de no ser de tipo WYSIWYG, se obtienen resultados de calidad brillante con poco esfuerzo y donde fundamentalmente el escritor se dedica a contar en su documento lo que le interesa sin preocuparse del formato, que para eso LaTeX se ocupa de “todo”.

POR JOSE MANUEL GONZÁLEZ VIDA



En esta penúltima entrega vamos a abordar en primer lugar la creación de tablas con LaTeX. Después trataremos la escritura matemática, donde podremos comprobar que LaTeX es infinitamente mejor y más potente que cualquier editor científico del tipo WYSIWYG. Posteriormente, en el próximo número, terminaremos nuestra serie sobre LaTeX viendo cómo incluir la bibliografía en nuestro documento (bien directamente o creándonos nuestra propia base de datos) y terminaremos con un ejemplo de un proyecto donde se usan todos los conceptos que hemos tratado en esta serie de artículos.

## ¿Cómo construir tablas con LaTeX?

Las tablas son construcciones alineadas que aunque no son demasiado fáciles de construir con LaTeX, sin embargo éste permite una enorme flexibilidad en su construcción y manipulación. Como siempre lo mejor será que vayamos aprendiendo viendo un ejemplo:

```
%Sintaxis de una tabla básica
\begin{tabular}[<I>Posición<I>]
```

```
{<I>FormatoColumnas<I>}
Fila1_Columna1 & Fila1_Columna2 &
& \dots & Fila1_ColumnaN \\
Fila2_Columna1 & Fila2_Columna2 &
& \dots & Fila2_ColumnaN \\
\dots & \dots & \dots & \dots \\
FilaM_Columna1 & FilaM_Columna2 &
& \dots & FilaM_ColumnaN \\
\end{tabular}
```

Como se puede comprobar, el entorno que LaTeX usa para construir tablas es el entorno *tabular* con determinados argumentos obligatorios y opcionales que vamos a ir viendo. La *Posición* es un argumento opcional e indica la posición vertical de la tabla respecto de la línea en la que se ha incluido. Puede tomar uno de los tres valores siguientes:

- *t*: la línea inferior de la tabla se alinea con la línea base de la línea actual.
- *b*: la línea superior de la tabla se alinea con la línea base de la línea actual.
- *c*: es la opción que se toma por defecto y alinea la tabla de manera centrada respecto de la línea base.

El argumento *FormatoColumnas* se refiere a la alineación de las columnas. Siempre debe aparecer una entrada por cada columna de la tabla. Se pueden incluir ciertos elementos opcionales (*separadores* para delimitar los bordes entre cada columna -los veremos en el siguiente ejemplo-). Las posibilidades son:

- *l*: alineación a la izquierda.
  - *r*: alineación a la derecha.
  - *c*: alineación centrada de la columna.
  - $p\{ancho\}$ : crea una columna de anchura *ancho*. El texto que se coloque dentro se trata como un párrafo (sin sangría inicial) con la línea superior alineada con las otras columnas.
- Los *Separadores* son elementos opcionales. Pueden tomarse entre los siguientes:
- $|$ : introduce una línea vertical.
  - $||$ : introduce dos líneas verticales.
  - $@\{texto\}$ : elimina el espacio entre columnas e inserta el *texto* señalado entre las columnas especificadas.
- Una vez vistos todos los argumentos del entorno *tabular* es el momento de pasar

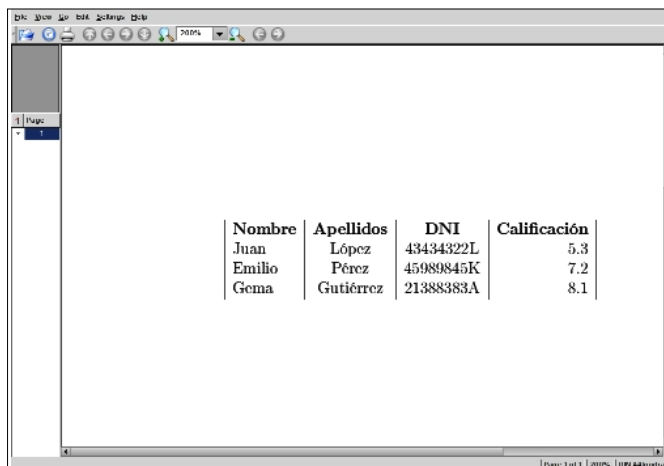


Figura 1: Un primer ejemplo de una tabla.

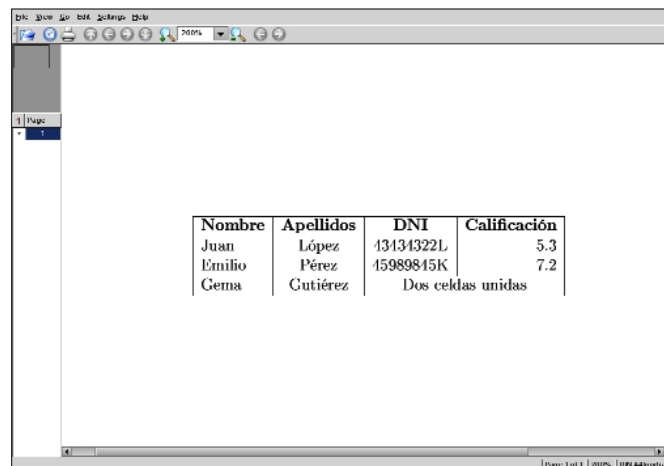


Figura 2: Una tabla con una "multicolumna".

a la práctica escribiendo en nuestro editor de texto el siguiente ejemplo:

```
01 \documentclass[a4paper,12pt]
02 {article}
03 \usepackage[latin1]{inputenc}
04 \usepackage[T1]{fontenc}
05 \usepackage[spanish]{babel}
06 \begin{document}
07 \begin{tabular}{|l|c|c|r|}
08 {\bf Nombre} & {\bf
09 Apellidos} & {\bf DNI} & {\bf
10 Calificación} \\
11 Juan & López & 43434322L &
12 5.3 \\
13 Emilio & Pérez & 45989845K &
14 7.2 \\
15 Gema & Gutiérrez & 21388383A &
16 8.1
17 \end{tabular}
18 \end{document}
```

El resultado de compilar el ejemplo anterior lo tenemos en la Figura 1. Podemos observar en el listado de nuestro ejemplo que no hemos usado los argumentos opcionales y que en el formato de las columnas hemos escogido separadores simples para las columnas verticales. Otro detalle importante que conviene observar es la utilización de los símbolos & y \ en las distintas filas de la tabla que

hemos construido. Con el símbolo & separamos las distintas columnas en una fila. Podemos dejar una casilla vacía colocando dos && o bien dejando un espacio en blanco entre ellos. Con las dos barras \ en indicamos a LaTeX que comienza una nueva fila.

Para separadores horizontales tenemos el comando \hline que dibuja una línea horizontal entre fila y fila. Debe aparecer justo antes de la primera fila o justo después de \ en si está entre dos filas. Dos comandos \hline seguidos dibujan dos líneas horizontales con un pequeño espacio entre ellas. \cline{n - m} dibuja una línea horizontal entre las columnas n y m. Si retocamos el ejemplo anterior cambiando la tabla usando el siguiente código:

```
\begin{tabular}{|l|c|c|r|}
\hline
{\bf Nombre} & {\bf Apellidos} &
{\bf DNI} & {\bf Calificación} \\
\hline
Juan & López & 43434322L &
5.3 \\
Emilio & Pérez & 45989845K &
7.2 \\
Gema & Gutiérrez & 21388383A &
8.1 \\
\hline
\end{tabular}
```

Obtenemos una tabla. Ahora me gustaría que probaseis a sustituir la tabla de nuestro ejemplo por la siguiente

```
\begin{tabular}{|l|c|c|r|}
\hline
{\bf Nombre} & {\bf Apellidos} &
{\bf DNI} & {\bf Calificación} \\
\hline
Juan & López & 43434322L &
5.3 \\
Emilio & Pérez & 45989845K &
7.2 \\
Gema & Gutiérrez &
\multicolumn{2}{c}{Dos celdas unidas} \\
\hline
\end{tabular}
```

El resultado, como podemos observar en la Figura 2, difiere del anterior en que hemos empleado el comando \multicolumn que se usa para unir varias columnas de una misma fila. \multicolumn usa como argumentos el número de columnas a unir, la alineación del texto a incluir en la "multicolumna" y, por último, el texto que deseamos incluir. Por último probad a sustituir la tabla anterior por la siguiente:

```
\begin{tabular}{|@{\bf}r}
Capítulo: } r@{\quad}
Temas: }c@{ - }c|}
\hline
1 & 1 & 5 \\
2 & 6 & 9 \\
3 & 10 & 14 \\
\hline
\end{tabular}
```

### Tabla 1. Algunas fórmulas matemáticas sencillas

Instrucción	Uso
$\_subíndice$	Texto en subíndice
$\^superíndice$	Texto en superíndice
$\sqrt[n]{radicando}$	Raíz enésima del radicando
$\frac{numerador}{denominador}$	Cociente entre el numerador y el denominador

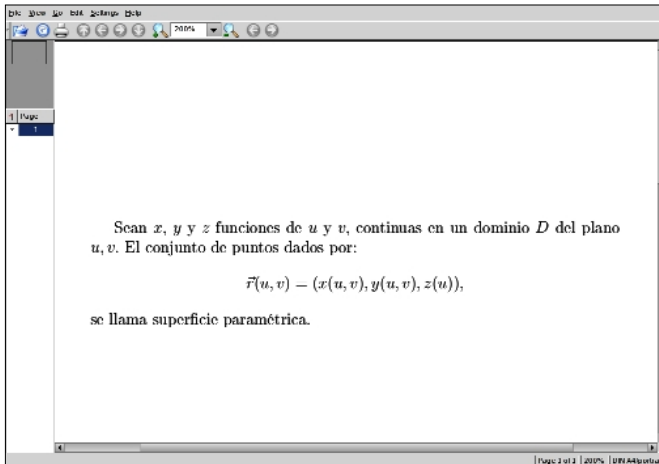


Figura 3: Un ejemplo de escritura en modo texto y en modo matemático.

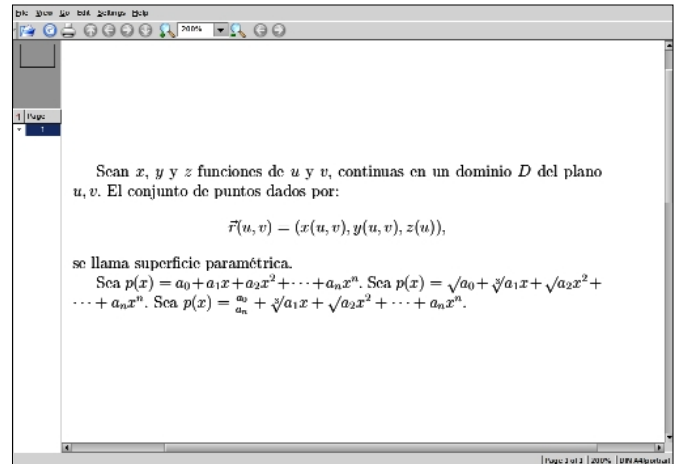


Figura 4: Un ejemplo de escritura matemática con fórmulas sencillas.

Fijaros en que en este ejemplo los separadores de las columnas son las palabras que hemos incluido en los argumentos del entorno `tabular`.

### Tablas como objetos flotantes

Recordareis que en el artículo del número anterior explicábamos el concepto de *objeto flotante* basándonos en la idea de que las figuras son objetos indivisibles y por tanto pueden “flotar” y ser reubicadas en nuestro documento (dentro de unas determinadas pautas) con el fin de que el aspecto global del documento cumpla nuestras expectativas. En este sentido las tablas no deberían (aunque hay excepciones) ser separadas en más de una página, por tanto su ubicación debe ser determinada por el compilador. Para tratar las tablas como objetos flotantes disponemos del entorno `table`. Su sintaxis es:

```
%Sintaxis del entorno table
\begin{table}
Entorno tabular
\caption{Pie de la tabla}
\label{Etiqueta}
\end{table}
```

Las tablas, al igual que las figuras, pueden llevar su etiqueta con el fin de referenciarlas en nuestro documento y también su pie, que nos servirá también para crear el índice de tablas.

### Modo Texto y Modo Matemático

Básicamente LaTeX distingue el texto matemático según se introduzca a lo

largo del párrafo que se está escribiendo o en párrafos separados. El texto de tipo matemático introducido en la misma línea de escritura (modo texto) ha de ser introducido entre los símbolos `$` y `$`. Para introducir el texto matemático en líneas separadas se suele hacer encerrándolo entre los símbolos `$$` y `$$` o bien entre los símbolos `\[` y `\]`. Introducimos el siguiente ejemplo dentro de un documento LaTeX con las cabeceras estándares que estamos usando y lo compilamos.

```
Sean $$, $$ y $$
funciones de $u$ y $v$,
continuas en un dominio
$D$ del plano $u, v$.
El conjunto de puntos
dados por:
$$
\vec{r}(u, v) = (x(u, v), y
(u, v), z(u)),
$$
se llama superficie
paramétrica.
```

El resultado debe ser parecido al que mostramos en la Figura 2. Observemos que los caracteres que usa LaTeX para la escritura en modo matemático son distintos a los hemos visto hasta ahora. Se trata de letras ligeramente cursivas y se emplea una fuente más redondeada.

### El Entorno Equation

Es frecuente, cuando se escriben textos científicos, hacer referencias a determinadas fórmulas que aparecen en el texto.

Para ello en LaTeX emplearemos el entorno `equation`. La sintaxis que tiene este entorno es:

```
%Sintaxis del entorno equation
\begin{equation}
\label{Etiqueta}
Fórmula
\end{equation}
```

Este entorno numera automáticamente las ecuaciones utilizando el contador denominado `equation`. De este modo podemos hacer referencia a las distintas fórmulas con los comandos `\ref{Etiqueta}` y `\pageref{Etiqueta}`. Además también disponemos del comando `\eqref{Etiqueta}` que, a diferencia de los anteriores, encierra el número asignado a la ecuación entre paréntesis.

A diferencia del modo texto, en modo matemático LaTeX presenta un comportamiento distinto, es decir, además de usar una fuente distinta y con letras de tipo itálico, no respeta los espacios entre palabras, sólo deja espacios cuando lo considera necesario. No se permite escribir vocales acentuadas ni la “ñ” en el modo matemático. LaTeX se ocupa de modo automático de elegir los tamaños adecuados de letra adecuados a diversas circunstancias (fracciones, superíndices o subíndices, etc.).

### Algunas Fórmulas Sencillas

En la Tabla 1 mostramos la forma de escribir super y subíndices, raíces desde cuadradas a cualquier orden, fracciones, etc. Una vez que veamos la sintaxis a usar en esta tabla, podemos escribir el

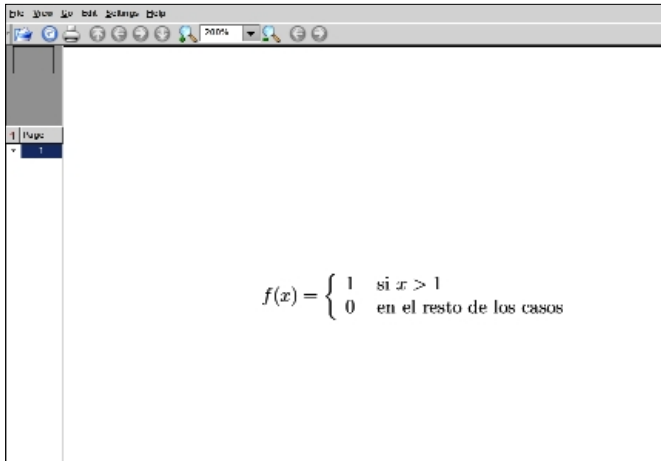


Figura 5: Una función definida a trozos.

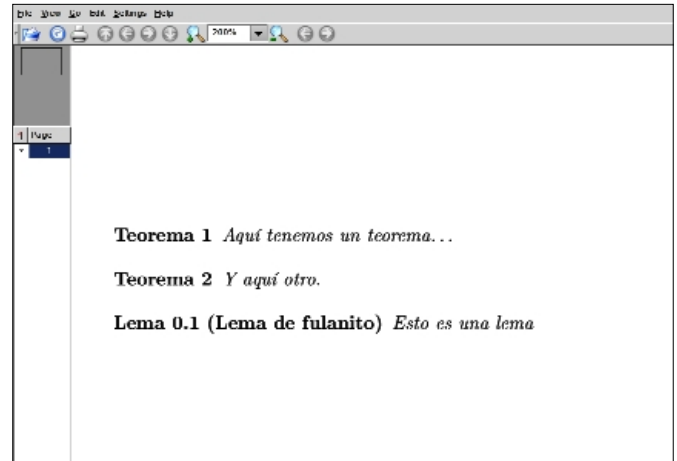


Figura 6: Uso de estructuras de tipo teorema.

siguiente ejemplo en el cuerpo de nuestro documento:

```
Sea $p(x)=a_0+a_1x+a_2x^2+\dots+a_nx^n$.
Sea $p(x)=\sqrt{a_0}+\sqrt[3]{a_1}x+\sqrt{a_2}x^2+\dots+a_nx^n$.
Sea $p(x)=\frac{a_0}{a_n}+\sqrt[3]{a_1}x+\sqrt{a_2}x^2+\dots+a_nx^n$.
```

Lo compilamos y visualizamos el resultado, que debe ser parecido al que mostramos en la Figura 4.

### El paquete *amsmath*

El paquete *amsmath* es una extensión para LaTeX que proporciona nuevas características para facilitar la escritura de fórmulas matemáticas y mejorar la calidad tipográfica del documento final. Como regla general incluiremos siempre este paquete en nuestros documentos cuando éstos contengan fórmulas y no distinguiremos en esta sección los comandos propios de LaTeX de los que nos proporciona el paquete *amsmath*.

Hemos visto antes que cuando escribimos texto en modo matemático LaTeX no respeta los espacios, y entonces ¿cómo incluir texto en nuestras fórmulas? La respuesta se halla en el comando `\text{texto a incluir}`. El *texto* que incluyamos dentro de este comando será tratado exactamente como los textos que hemos visto hasta ahora.

También tenemos en LaTeX otra serie de comandos que nos proporcionan la posibilidad de dejar espacios de distinto tamaño en nuestras fórmulas en modo

matemático. Los podemos encontrar en la Tabla 2.

### Construcciones Especiales en Modo Matemático

Como estaréis comprobando el modo matemático da mucho de sí en LaTeX. En este apartado vamos a aprender a construir matrices y lo que se denominan funciones definidas a trozos. Si imaginamos una matriz en matemáticas podemos pensar que para escribirla con LaTeX hay que usar el entorno *tabular* que vimos antes pero este entorno es válido sólo en modo texto. El entorno equivalente en modo matemático es *array*. Su sintaxis es parecida a la del entorno *tabular*. Lo vemos en el siguiente ejemplo:

```
01 \documentclass[a4paper,
02 12pt]{article}
03 \usepackage[latin1]{inputenc}
04 \usepackage[T1]{fontenc}
05 \usepackage[spanish]{babel}
06 \usepackage{amsmath}
07 \begin{document}
08 \left(
09 \begin{array}{ccc}
10 1 & 2 & 3 \\
11 4 & 5 & 6 \\
12 7 & 8 & 9
13 \end{array}
14 \right)
15 $$
16 \end{document}
```

El resultado obtenido al compilar, hablando sólo en términos de formato, es una tabla (como las del entorno *tabular*) precedida y terminada por unos paréntesis

que se adaptan al tamaño de la misma. Pues bien, el entorno *array* permite ir precedido por el comando `\left` y terminado por `\right`. Observemos que en nuestro ejemplo hemos escrito el símbolo ( justo después del comando `\left` y ) justo después del comando `\right`. Son estos comandos los que hacen que el símbolo que colocamos justo después adapte el tamaño del *array*. Además permiten estructurar un poco más especiales. Escribid el siguiente código en el cuerpo del ejemplo anterior:

```
$$
f(x)=
\left(
\begin{array}{ll}
1 & \mbox{ si } x>1 \\
0 & \mbox{ en el resto de }
\text{ los casos}
\end{array}
\right)
$$
```

El resultado de compilar el ejemplo anterior lo tenemos en la Figura 5. Se trata de una función definida a trozos, que de hecho, no es más que un array de dos columnas que va precedido de un símbolo de llave ( ) y que no se cierra con otra llave ( fijaros que para no cerrar el *array* con un símbolo usamos `\right` ). Además hemos usado otro nuevo comando `\mbox` que nos sirve al igual que el `\text` para poder escribir texto dentro del modo matemático.

### Simbología Matemática

La simbología matemática en LaTeX es increíblemente extensa y abarca varios

**Tabla 2. Espaciados en Modo Matemático**

Comando	Abreviatura	Espacio proporcionado
<code>\thinspace</code>	<code>\,</code>	pequeño
<code>\medspace</code>	<code>\:</code>	mediano
<code>\thickspace</code>	<code>\;</code>	grande
<code>\negthinspace</code>	<code>\!</code>	espacio negativo pequeño
<code>\negmedspace</code>		espacio negativo mediano
<code>\negthickspace</code>		espacio negativo grande

miles de símbolos que van desde las clásicas letras griegas, todo tipo de flechas, operadores binarios, funciones matemáticas y un largo etcétera que por razones de espacio no podemos incluir aquí pero que podemos encontrar en el libro “Una Descripción de LaTeX2e” realizado por Tomás Bautista, sobre el trabajo inicial de Hubert Partl, Irene Hyna, Elisabeth Schlegl y Tobias Oetiker y del que hemos puesto un enlace a una versión en PDF en el punto [3] de la bibliografía de este artículo. Además, a modo de ejemplo, os he incluido en la Figura 7 cómo se escriben todas las letras clásicas griegas con LaTeX.

## Teoremas y Demostraciones

Otros elementos comunes dentro de artículos de investigación y libros de contenido matemático son los teoremas y demostraciones. Atendiendo al sistema de numeración llamaremos entornos de tipo teorema a estructuras de tipo teorema, proposición, lema, corolario, definición, conjetura, etc. En LaTeX tenemos el comando `\newtheorem` que nos permitirá definir en nuestro documento tantos entornos de tipo teorema como necesitemos. Como siempre, lo mejor es comprender su manejo con un ejemplo:

```
01 \documentclass[a4paper,12pt]{article}
02 \usepackage[latin1]{inputenc}
03 \usepackage[T1]{fontenc}
04 \usepackage[spanish]{babel}
05 \begin{document}
06 \newtheorem{teorema}{Teorema}
07
\newtheorem{lema}{Lema}[section]
08 \begin{teorema}
09 Aquí tenemos un teorema\dots
10 \end{teorema}
11 \begin{teorema}
12 Y aquí otro.
```

```
13 \end{teorema}
14 \begin{lema}[Lema de
fulanito]
15 Esto es una lema
16 \end{lema}
17 \end{document}
```

El resultado de compilar el texto anterior lo mostramos en la Figura 6. ¿Qué hemos conseguido? Pues mediante el comando `\newtheorem` hemos creado nuestros propios entornos y les hemos asignado un nombre de entorno y un contador propio. Me explico, mediante la línea `\newtheorem{teorema}{Teorema}` esta-

Letras minúsculas					
$\alpha$	<code>\alpha</code>	$\beta$	<code>\beta</code>	$\gamma$	<code>\gamma</code>
$\delta$	<code>\delta</code>	$\epsilon$	<code>\epsilon</code>	$\zeta$	<code>\zeta</code>
$\eta$	<code>\eta</code>	$\theta$	<code>\theta</code>	$\iota$	<code>\iota</code>
$\kappa$	<code>\kappa</code>	$\lambda$	<code>\lambda</code>	$\mu$	<code>\mu</code>
$\nu$	<code>\nu</code>	$\xi$	<code>\xi</code>	$\omicron$	<code>\omicron</code>
$\pi$	<code>\pi</code>	$\rho$	<code>\rho</code>	$\sigma$	<code>\sigma</code>
$\tau$	<code>\tau</code>	$\upsilon$	<code>\upsilon</code>	$\phi$	<code>\phi</code>
$\chi$	<code>\chi</code>	$\psi$	<code>\psi</code>	$\omega$	<code>\omega</code>

Letras mayúsculas					
$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>

**Figura 7: Las letras griegas con LaTeX.**

mos creando un nuevo entorno que se llama **teorema** de tal manera que cada vez que usamos un entorno de este tipo actualizamos un contador que va desde el 1,2,... y creamos un texto que comienza con **Teorema x** seguido del enunciado que queremos escribir dentro de este entorno (fijaros en el ejemplo anterior). Pero a veces nos puede gustar más que el contador de un entorno de este tipo vaya ligado a la sección donde se encuentra, con el fin de que, al referenciarlo, el lector sepa en qué sección tiene que buscarlo. ¿Cómo se hace? Pues muy fácil, en el ejemplo anterior hemos creado un entorno con nombre **lema** y cuyo contador está referenciado a las secciones (mirad en el ejemplo cómo ha resultado). El contador resultante lleva en primer lugar el número de la sección actual y separado con un punto el conta-

dor propio del entorno que hemos creado.

Para las demostraciones podemos usar el paquete `amsmath` que contiene el entorno `proof`. Este entorno produce un encabezamiento con la palabra **Proof**. y produce un símbolo en forma de cuadrado al final. Para redefinir el encabezamiento ejecutaremos el siguiente código dentro de nuestro documento (da igual si se hace en el preámbulo o en el cuerpo del mismo):

```
\renewcommand*{\proofname}{Demostración}
```

En el siguiente ejemplo os muestro cómo se usa.

```
01 \documentclass[a4paper,12pt]{article}
02 \usepackage[latin1]{inputenc}
03 \usepackage[T1]{fontenc}
04 \usepackage[spanish]{babel}
05 \begin{document}
06 \renewcommand*{\proofname}{Demostración}
07 \begin{proof}
08 Prueba del resultado\dots
09 \end{proof}
10 \end{document}
```

## Conclusión

Con lo visto hasta ahora sois ya unos casi LaTeXpertos, pero remataremos nuestros conocimientos en la próxima entrega donde podremos ver a LaTeX en todo su esplendor, creando un proyecto con capítulos, índices, figuras, tablas, bibliografía y ¡todo de manera automática! Además os pondremos en nuestra página <http://www.linux-magazine.com>. es los listados de este proyecto para que los tengáis de referencia para vuestros propios trabajos. ■

## RECURSOS

- [1] El libro de LaTeX. B. Cascales y otros. Ed. Prentice Hall. 2003.
- [2] LaTeX, a document preparation system. L. Lamport. 2ª Ed. Ed. Addison-Wesley. 1994.
- [3] “Una Descripción de LaTeX2e”. Tomás Bautista. <http://www.iiuma.ulpgc.es/users/bautista/other/tex/ldesc2e/misc/ldesc2e.pdf>