

# Ejemplo de documento en LaTeX

Marta Andrés

David Charte

24 de octubre de 2016

## 1. Introducción

Este documento PDF ha sido generado al compilar la `demo ejemplo.tex` que viene incluida en el repositorio de preparación para primer curso. En las siguientes secciones se exploran algunas de las opciones básicas de L<sup>A</sup>T<sub>E</sub>X para formateo de texto e inclusión de otros elementos, como ecuaciones, tablas y figuras.

## 2. Formato

Dar formato al texto en L<sup>A</sup>T<sub>E</sub>X es relativamente sencillo. Basta con usar algunos comandos usuales para **resaltar texto en negrita**, tal vez *añadir algunas palabras en cursiva*, incluso código en monoespaciada. Para añadir un bloque de código podemos usar el entorno `verbatim`.

```
def count_words string
  string.split(" ").length
end
```

## 3. Matemáticas

### 3.1. El entorno matemático

El código que escribes en L<sup>A</sup>T<sub>E</sub>X regularmente sólo acepta texto normal, no podemos escribir ecuaciones directamente sobre él. Sin embargo, hay comandos y entornos que activan el **entorno matemático**. Dentro de él, podrás dibujar símbolos matemáticos mediante comandos especiales.

### 3.2. Ecuaciones en línea

El entorno matemático en línea (es decir, integrado en el párrafo de texto) se activa y termina con el signo del dólar `$`. Veamos un ejemplo:  $e^{i\pi} = 1$ .

### 3.3. Ecuaciones en bloque

A las ecuaciones que ocupan su propio renglón y van separadas del texto se les suele llamar *en bloque*. Se pueden englobar en el entorno `equation` o bien

entre `\[` y `\]`. La diferencia entre estas dos opciones es que la primera numerará la ecuación

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \quad (1)$$

y la segunda no (un resultado similar se consigue con el entorno `equation*`):

$$f((x, y)) = (-y, x)$$

### 3.4. Símbolos

En el modo matemático podemos escribir variedad de símbolos, por ejemplo las letras griegas  $\alpha, \beta, \gamma$  y las demás. Para una referencia más completa, visita este libro de Wikibooks.

### 3.5. Más ecuaciones

$$\delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{en otro caso} \end{cases}$$
$$Ax = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

## 4. Figuras, tablas y otros

A menudo se quiere incluir una imagen en el documento. Con el comando `includegraphics` se añade la imagen al documento, pero para que quede bien posicionada la ponemos en un entorno `figure`. Además, le podemos añadir un pie de foto que numera automáticamente la imagen con el comando `caption`.



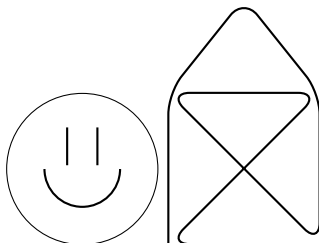
Figura 1: Los gatitos siempre son adorables.

También puedes querer hacer una tabla, como por ejemplo la tabla de sumar en  $\mathbb{Z}_2$ . Para ello se usa el entorno `tabular`, aunque hay otros entornos más potentes que se pueden usar para hacer tablas más complejas.

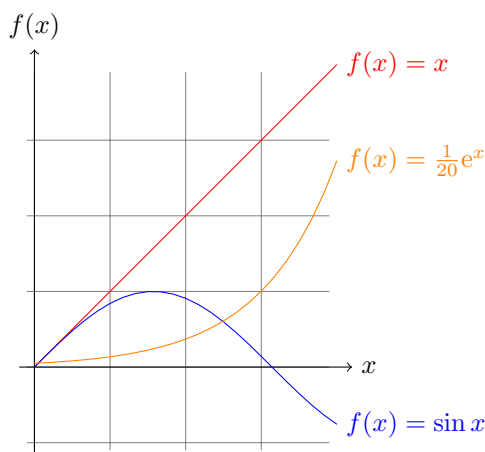
+	0	1
0	0	1
1	1	0

Tabla 1: Tabla de sumar de  $\mathbb{Z}_2$ .

A veces también nos interesa crear nuestro propio diagrama cuando es sencillo, y para eso se usa el paquete de gráficos `tikz`. Si conocemos las coordenadas clave del diagrama, con el entorno `tikzpicture` podemos dibujar cosas simples como una cara sonriente o la solución al problema de la casa que se dibuja sin levantar el boli del papel (ejemplo tomado del Wikibook de  $\text{\LaTeX}$ ):



Podemos usar el mismo entorno para dibujar gráficos de funciones (ejemplo tomado del Wikibook de  $\text{\LaTeX}$ ):



#### 4.1. Dia

Para ahorrar tiempo a la hora de crear diagramas, podemos usar un software llamado **Dia** que nos permite crear nuestro diagrama de forma visual y exportarlo al formato de Tikz. Al exportarlo obtendremos un documento con el diagrama en sintaxis Tikz para poder modificar lo que necesitemos (por ejemplo, poner un símbolo matemático en el diagrama). También nos permite hacer

diagramas UML y un montón de cosas más. Aclaro que cuando el diagrama tiene colores la exportación a  $\text{\LaTeX}$ no suele ir muy bien. Para añadir el diagrama creado por Dia a nuestro documento podemos incluirlo en el comando `figure`, tal y como se ha hecho en la Figura 2.

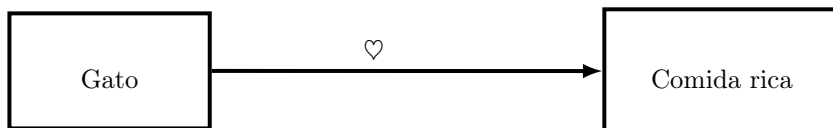


Figura 2: Diagrama hecho con Dia

## 4.2. Subfiguras

Para poner varias imágenes en una misma figura, usamos el paquete `subfigure`. Para poder dos imágenes una al lado de la otra debemos usar `mbox`.

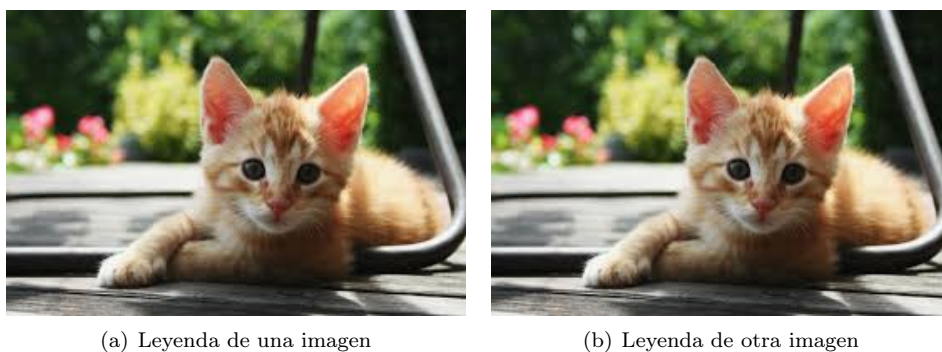


Figura 3: miau

## 5. Incluir código

Para incluir código en nuestro documento podemos usar el paquete `minted`. Para instalarlo en Linux, usamos el siguiente comando:

```

----- Instalación de minted en Arch Linux -----
$ sudo pacman -S minted
$ sudo pacman -S python-pygments
  
```

```

----- Instalación de minted en Ubuntu -----
$ sudo apt-get install texlive-latex-extra
$ sudo apt-get install python-pygments
  
```

Para ver todos los lenguajes que soporta podemos ejecutar el siguiente comando:

```

----- Lenguajes soportados por minted -----
$ pygmentize -L lexers
  
```

Otra opción es importar directamente a nuestro PDF el código al completo de un fichero fuente externo. Esto es útil ya que si modificamos nuestro código, al compilar el fichero `.tex`, automáticamente tendremos el código actualizado, evitando tener que ir modificando el código en sitios diferentes. Para ello, introducimos en la cabecera el siguiente fragmento de código:

```
_____ Importar ficheros de código en LaTeX _____  
\newmintedfile[nombreDeLaEtiqueta]{python}{  
  linenos,  
  numbersep=5pt,  
  gobble=0,  
  frame=lines,  
  framesep=2mm,  
}
```

En este caso, lo haremos para ficheros con código escrito en Python, e importaremos el fichero de la siguiente manera:

```
_____ Importar ficheros de código en LaTeX _____  
\mycpp[label="Nombre del frame"]{prueba.py}
```

Usemos la opción que usemos, para compilar el fichero  $\text{\LaTeX}$ , tenemos que usar la siguiente orden en la consola:

```
_____ Lenguajes soportados por minted _____  
$ pdflatex -shell-escape fichero.tex
```