

# Tema 0: Introducción

## La terminal de GNU/Linux

Este material ha sido desarrollado en su totalidad por Román Ginés Martínez Ferrández ([rgmf@riseup.net](mailto:rgmf@riseup.net)) salvo referencias al pie de página.

Todas las imágenes utilizadas son de Dominio Público a menos que se diga lo contrario.



Creative Commons Reconocimiento – NoComercial - CompartirIgual  
CC by-nc-sa

## Sumario

1. ¿Qué es la Terminal?.....	1
2. Sintaxis general.....	1
3. Comandos básicos.....	1
3.1. cd.....	1
3.2. pwd.....	2
3.3. ls.....	2
3.4. cat, more.....	2
3.5. mkdir.....	3
3.6. rm.....	3
3.7. cp.....	4
3.8. mv.....	4

## 1. ¿Qué es la Terminal?

El emulador de terminal o simplemente **terminal**, **consola** o **shell** es una interfaz en línea de comandos o CLI (*Command Line Interface*) que nos permite usar un ordenador y comunicarnos con él por medio de **órdenes o comandos**.

Estas fueron las primeras interfaces, luego aparecieron las interface gráficas o GUI (*Graphical User Interface*). No obstante, hoy en día las siguen usando administradores de sistema, programadores y usuarios avanzados por la gran potencia que ofrecen.

La terminal se encuentra en todos los sistemas operativos: Windows, MacOS, GNU/Linux, Android, iOS, etc.

### Nota 1

Los comandos u órdenes son en realidad programas que llevan a cabo una serie de instrucciones.

### Nota 2

Aunque no es exactamente lo mismo un emulador de terminal, un terminal, una consola o un *shell*, se suelen utilizar como sinónimos.

Lo que nosotros vamos a utilizar es en realidad un emulador de terminal o *shell*, aunque lo vamos a llamar indistintamente como terminal, consola o *shell*.

## 2. Sintaxis general

Los comandos, en general, siguen la siguiente sintaxis:

```
comando [opciones] [argumentos]
```

donde:

- `comando` es el nombre de la orden,
- `opciones` son determinadas opciones que se pueden aplicar sobre el comando y que cambian su comportamiento, y
- `argumentos` son datos que se le pasan al comando.

No todos los comandos tienen opciones y/o argumentos.

## 3. Comandos básicos

### 3.1. cd

El comando `cd` (*change directory*) se utiliza, como su nombre indica, para cambiar a un directorio.

#### Directorio

Un directorio es una carpeta. Aunque son sinónimos en el contexto del *shell* se suele utilizar el término directorio.

Se usa de la siguiente manera:

```
cd <ruta absoluta o relativa>
```

Por ejemplo:

- `cd /home/roman` cambia al directorio `/home/roman`

### 3.2. pwd

El comando `pwd` (*Path WorD*) Imprime por pantalla la ruta absoluta donde nos encontramos en estos momentos. Aunque tiene algunas opciones (muy pocas) nosotros lo vamos a utilizar tal cual, sin opciones ni argumentos.

### 3.3. ls

El comando `ls` (*LiSt*) imprime por pantalla los archivos y los directorios que hay donde estamos.

Se utiliza de la siguiente manera:

```
ls [opciones]
```

Por ejemplo:

- `ls` imprime por pantalla con un formato estándar
- `ls -l` con la opción `-l` se imprime por pantalla los archivos y los directorios con información adicional como: usuario, fecha última actualización, permisos, etc.
- `ls -t` ordena los archivos y los directorios por fecha de última actualización.
- `ls -l -t` se aplican las opciones `-l` y `-t`.
- `ls -lt` se aplican las opciones `-l` y `-t`.

#### Aclaraciones

Como ves en los ejemplos de arriba las opciones constan de un guión seguido de una letra. Además se pueden concatenar opciones como ves en los dos últimos ejemplos.

### 3.4. cat, more

Los comandos `cat` (*conCAT*) y `more` imprimen por pantalla el contenido de un fichero de texto.

Estos comandos requieren como argumento la ruta absoluta o relativa del fichero de texto:

```
cat <ruta al fichero de texto>
more <ruta al fichero de texto>
```

Algunos ejemplos:

- `cat /home/roman/informatica.txt` Muestra el contenido del fichero `informatica.txt` que está dentro de `/home/roman`.
- `more ../informatica.txt` Muestra el contenido del fichero `informatica.txt` que está en el directorio de arriba.
- `more informatica.txt` Muestra el contenido del fichero `informatica.txt` que está en el directorio de trabajo actual.

### 3.5. mkdir

El comando `mkdir` (*MaKe DIRectory*) crea un directorio en el lugar especificado. Este comando necesita, al menos, un argumento: el nombre del nuevo directorio.

Este comando lo usamos de la siguiente manera:

```
mkdir <dir1> <dir2> <dir3> ... <dirN>
```

Algunos ejemplos:

- `mkdir /home/roman/nuevo` Crea el directorio `nuevo` dentro de `/home/roman`.
- `mkdir uno dos tres` Crea los directorios `uno`, `dos` y `tres` en el directorio de trabajo actual.
- `mkdir Documentos/nuevo1 nuevo2` Crea los directorios `nuevo1` dentro de `Documentos` y `nuevo2` dentro del directorio de trabajo.

### 3.6. rm

El comando `rm` (*ReMove*) se utiliza para eliminar ficheros y directorios. Se utiliza de la siguiente manera para eliminar ficheros:

```
rm <ruta al fichero a eliminar>
```

Y de la siguiente manera (con la opción `-r`) para eliminar directorios (aunque también borraría ficheros):

```
rm -r <ruta al fichero o directorio a eliminar>
```

Ejemplos:

- `rm Documentos/informatica.txt` Borra el fichero `informatica.txt` que hay en `Documentos`.

- `rm -r /home/roman/Documentos` Borra el directorio Documentos y todo lo que haya dentro (es peligroso y hay que utilizarlo con cuidado).
- `rm uno.txt dos.txt tres.txt` Borra los tres ficheros.
- `rm -r uno.txt dir1 dir2` Borra el fichero y los tres directorios.

### 3.7. cp

El comando `cp` (*CoPy*) se utiliza para copiar ficheros y directorios. Como en el caso de `rm` se utiliza la opción `-r` para copiar directorios y sin opción para copiar ficheros. Además, se necesitan dos argumentos:

- origen: ruta al fichero o directorio que se quiere copiar.
- destino: ruta al fichero o directorio donde se quiere pegar.

```
cp <origen> <destino>
cp -r <origen> <destino>
```

Algunos ejemplos:

- `cp /home/roman/infor.txt /home/roman/Documentos/infor.txt`  
Copia el fichero `infor.txt` que está en `/home/roman` dentro de `/home/roman/Documentos` con el mismo nombre.
- `cp Documentos/avatar.jpg Escritorio/avatar.jpg`  
Copia la imagen `avatar.jpg` que está dentro de Documentos en la carpeta Escritorio con el mismo nombre.
- `cp snoopdog.mp3 Musica/snoopdog.mp3`  
Copia `snoopdog.mp3` que está en el directorio de trabajo dentro de la carpeta Musica con el mismo nombre.
- `cp inform.txt Documentos/informatica.txt`  
Copia el fichero `inform.txt` que hay en el directorio de trabajo dentro de la carpeta Documentos con el nombre `informatica.txt` (se le ha cambiado el nombre al fichero).
- `cp snoopdog.mp3 Musica/.`  
Forma especial de copiar dentro de un directorio un fichero con el mismo nombre: mediante el `“.”` al final como ves.

### 3.8. mv

El comando `mv` (*MoVe*) se utiliza para mover archivos y directorios de un sitio a otro. Su sintaxis es muy sencilla:

```
mv <origen> <destino>
```

donde tanto origen como destino son rutas: la ruta del fichero que quiero mover y la ruta del lugar donde lo quiero mover, respectivamente.

Algunos ejemplos:

- `mv foto.png Imagenes/foto.png`

Mueve la imagne foto.png a la carpeta Imagenes con el mismo nombre.

- `mv foto.png Imagenes/.`

Igual que el comando anterior pero usamos el comodín “.” para indicar que la queremos mover con el mismo nombre.

- `mv foto.png Imagenes/foto_coche.png`

Movemos la imagen foto.png dentro de Imagenes cambiando el nombre a foto\_coche.png.