

# La Terminal

La terminal (a veces llamada también "consola" o "command line interface"(cli)) es la principal forma de interactuar que vamos a tener con nuestro servidor, especialmente si éste no tiene una interfaz gráfica.

En realidad, todo corre a través de la consola; la interfaz gráfica simplemente nos genera un entorno más "amigable" y, bueno, gráfico, para interactuar con ella, en lugar de "comandos de texto". (Envía dichos comandos de texto por nosotros)

## Tutorial: conceptos básicos de la consola

Si estás acá es porque ya estamos manejándonos usando la consola como principal forma de interactuar con el server. No paniquees! Es mucho más simple de lo que parece. Ante todo, si querés un tutorial un poco más completo, aunque en inglés, [recomiendo este de acá](#). Si preferís algo más superficial, sigamos!

### Qué es la consola?

La consola, simplemente, es un programa que nos permite interactuar con la computadora a través de comandos de texto. Para abrir una, en Linux o Mac, simplemente tenés que buscar el programa "Terminal". En caso de que tengas Windows, la "consola" que tienen ellos se va a llamar la Command Prompt (en español, si no me equivoco, "Símbolo del Sistema", y en ambos idiomas se encuentra como "CMD", también).

**IMPORTANTE!** En este tutorial, todo lo que veamos va a aplicar a consolas de Linux o macOS (sistemas operativos basados en "UNIX", un sistema operativo abuelito suyo). Si querés seguir el tutorial desde Windows, te recomiendo que busques el "WSL" (Windows Subsystem for Linux), o que lo hagas ya desde una conexión "ssh" con tu servidor.

Si te conectaste por [ssh], felicitaciones! Ya estás usando una terminal.

### Lo fundamental

Cuando trabajamos en una terminal (sea con una [conexión remota mediante SSH], o abriendo un "emulador de terminal" (a veces llamado simplemente terminal) o en la consola misma de la

computadora, estamos esencialmente usando un programa que espera que corramos "comandos" indicándole qué hacer. Este programa funciona de forma similar al navegador "explorer" de Windows o el "finder" de mac, en un sentido: Los comandos que corramos, va a correrlos en el contexto de una ubicación específica. Cuando recién lo abrimos, esa ubicación suele ser el directorio "home" de nuestro usuario. Por ejemplo, una sesión recién abierta se podría ver así:

```
fulano@waystone-inn:~$
```

Parecen cosas al azar, pero en realidad, la consola nos muestra toda información importante para tener en cuenta:

- Al comienzo, y a la izquierda de la @, podemos ver qué usuario está usando la consola (con quién nos logueamos). Este usuario, por ejemplo, se llama fulano.
- Del otro lado de la @ (que en inglés significa "at", es decir, "en"), tenemos el nombre de la computadora a la que estamos conectados, en este caso waystone-inn. Si leemos ambas secciones juntas, se podrían traducir a "fulano en la computadora waystone-inn".
- Después de los dos puntos, tenemos la dirección en la que estamos trabajando. El símbolo ~ simboliza nuestro "home", o carpeta "hogar" de nuestro usuario. Esta carpeta es sólo nuestra, y ningún otro usuario (salvo los administradores) puede acceder a ella a menos que decidamos permitirlo.
- Finalmente, el símbolo \$ representa que esta sesión es de un usuario común, que no es un administrador (en realidad, que no estamos actuando a través del usuario especial root, que es un usuario cuyos todos comandos son corridos como de administración. En ese caso, veríamos un # en su lugar).
- Finalmente finalmente, vemos el cursor de texto, esperando que escribamos algo.

Qué pasaría si corremos un comando? Probemos con "pwd", que como su nombre indica (**p**ath to **w**orking **d**irectory), nos muestra cuál es el "camino" hacia el directorio de la carpeta en la que estamos trabajando siguiendo la [[estructura jerárquica de directorios]] de Linux. Ingresamos pwd, y tocamos "enter" para enviar el comando, y obtenemos esto:

```
fulano@waystone-inn:~$ pwd
/home/fulano
fulano@waystone-inn:~$
```

Después de tocar "enter", la computadora nos responde con la dirección al directorio actual (/home/fulano), es decir que desde la carpeta base, conocida como /, entraríamos a la carpeta home, y dentro de ésta estaría la carpeta fulano), y en otra línea nos figura la misma info que antes, simbolizando que nuevamente tenemos control sobre la computadora (y nuevamente está

esperando una indicación o comando).

Es importante notar que cuando ingresamos "pwd", no tuvimos que aclarar a dónde era esto, ya que la consola dio por sentado que nos referíamos al directorio en el que nos encontrábamos. Veamos un ejemplo distinto. Si la consola nos mostrase, después de los dos punto, esto:

```
fulano@waystone-inn: /var/log/apt$
```

Nos estaría indicando que no estamos trabajando en el directorio "home" de nuestro usuario, sino en el directorio que se encuentra en la ruta `/var/log/apt`. Si corriésemos el comando `pwd`, por más redundante que sea, nos mostraría como resultado que, en efecto, ese es el "camino" hacia la carpeta en la que estamos trabajando, y que la consola da por sentado que estamos corriendo el comando *desde esa ubicación en los directorios del sistema*:

```
fulano@waystone-inn: /var/log/apt$ pwd
/var/log/apt
fulano@waystone-inn: /var/log/apt$
```

De la misma forma, y volviendo al directorio "home" (que figura como un `~` en nuestra consola, para abreviar), si usásemos el comando `touch`, que sirve para crear un archivo vacío, y se utiliza escribiendo `touch <nombre del archivo>`, el archivo creado lo encontraríamos en el actual directorio:

```
fulano@waystone-inn: ~$ touch archivo
fulano@waystone-inn: ~$
```

Un par de cosas para notar:

- Los archivos en Linux no *necesitan* llevar una extensión. Por lo general, si no indican nada, suelen ser archivos de texto, aunque a veces son binarios también. No nos importa mucho esa distinción, sí nos importa saber que **muchos programas usan la extensión para saber cómo tratar al archivo**, así que, aunque no sea necesaria (como en el ejemplo), es mejor ponerla por las dudas (en este caso, podría haber agregado `.txt` al final, por ejemplo).
- En Linux, si un comando no te dice nada cuando lo corrés, y asumiendo que es un comando que no nos da información específicamente (como `pwd`), \*significa que el comando se corrió con éxito`.

Pero, bancá un segundo. Cómo sabemos que se creó el archivo? Bueno, es que el comando `touch` sólo crea el archivo, y siguiendo la filosofía Unix de *crear herramientas que sólo hagan una cosa y la hagan bien*, no hace más nada. Ni siquiera nos muestra el archivo. Para verlo, podemos usar el comando `ls`, que nos **lista** los archivos y directorios dentro del directorio en el que nos encontramos:

```
fulano@waystone-inn:~$ touch archivo
fulano@waystone-inn:~$ ls
archivo
fulano@waystone-inn:~$ █
```

Podemos ver, ahora, que el archivo está en la carpeta en la que estamos (y, de paso, que es el único archivo presente).

Más abajo vamos a ver, entonces, varios comandos que podemos utilizar en la consola para navegarla y modificarla, creando archivos, copiándolos, cambiándoles el nombre y más.

## Navegando

Comando `cd`

Comando `ls`

## Creando archivos y directorios

Comando `mkdir`

Comando `touch`

## Copiando, moviendo y eliminando

Comando `mv`

Comando `cp`

Comando `rm`

## Editando archivos de texto

Comando `nano`

---

Revision #4

Created 1 November 2025 21:46:58 by Kvothe

Updated 10 November 2025 03:48:09 by Kvothe