

CLI

Interfaz de la línea de comandos



Contenido

INTRODUCCIÓN	3
¿Por qué una terminal?	4
Conceptos básicos.....	4
Interfaz de usuario	5
Intérprete de comandos.....	5
Terminal vs emulador de terminal	6
¿Qué es un comando?.....	6
¿Qué es un script?	8

INTRODUCCIÓN

Las interfaces de línea de comandos (CLI), también conocida como consola, terminal o símbolo del sistema, era la única forma de interactuar con una computadora antes que aparecieran las interfaces gráficas de usuario (GUI).

Estas interfaces de usuario fueron muy populares en la década de los 70 y 80, ya que los sistemas de la época carecían de interfaces gráficas y era el sistema habitual para comunicarse con un dispositivo. Con el tiempo, las interfaces gráficas de usuario ganaron popularidad y relegaron a las interfaces de línea de comandos a un segundo plano, aunque nunca terminaron de perder la popularidad que las caracterizaba, sobre todo entre los usuarios más técnicos.

Windows, Linux, macOS, cualquiera que sea el sistema operativo que utilices, es simplemente una representación visual de la computadora. Esto se conoce como interfaz gráfica de usuario (GUI).

Si quitas el explorador de archivos, el escritorio, los iconos y todos los demás gráficos, el sistema operativo se quedará sólo con la interfaz de línea de comandos. En lugar de arrastrar, soltar, señalar y hacer clic, tendrías que escribir todas las instrucciones. Esa es la única diferencia entre las CLIs y las GUIs.

Realmente puedes acceder a todos los elementos del sistema operativo de tu computadora a través de la línea de comando, puedes realizar tareas como:

- Navegar por las carpetas de tu computadora.
- Manipular archivos (crear, editar, copiar, mover y eliminar).
- Conectarte a servidores remotos.
- Crear scripts que te ayuden con tareas del sistema operativo.

¿Por qué una terminal?

Aunque una interfaz gráfica de usuario (GUI) puede ser más cómoda y agradable para el usuario, las razones para preferir una interfaz de línea de comandos son muchas:

- Es más rápido crear un programa para línea de comandos.
- Suele ser más potente en cuanto a personalización de opciones.
- Permite creación de scripts para automatizar tareas repetitivas.
- Suele ser una opción más universal entre sistemas.
- Suele ser instantáneo, al no tener que cargar pantallas gráficas.

Es por ello que en el mundo de la informática y del desarrollo en particular, es una vertiente muy utilizada aún en nuestros días, que no suele ser comprendida por personas externas al sector, que lo consideran un «atraso» al no conocer sus ventajas y sólo conocer sus desventajas.

No obstante, el panorama de línea de comandos ha mejorado muchísimo desde estas primeras décadas y hoy en día existen multitud de opciones para convertir una terminal en una herramienta de trabajo muy productiva.

Conceptos básicos

Hasta ahora, para simplificar, hemos utilizado una gran cantidad de términos que no son exactamente lo que parecen, por lo que antes de continuar, vamos a dedicar unos minutos a explicarlos, ya que los utilizaremos bastante:

- **Interfaz de usuario:** Medio (genérico) por el cual se comunica un usuario y una máquina. Abreviada UI.
- **Interfaz de línea de comandos:** Interfaz de usuario (UI) basado en texto. Abreviada como CLI.
- **Interfaz gráfica de usuario:** Interfaz de usuario (UI) basado en paneles gráficos o táctiles. Abreviada como GUI.
- **Intérprete de comandos:** Software para gestionar comandos del usuario y sus respuestas. Abreviado como Shell.

- **Terminal o Consola:** Dispositivo utilizado antiguamente para comunicarse con un sistema.
- **Emulador de terminal:** Software que simula ser una terminal, generalmente una ventana donde hay una interfaz CLI.
- **Comando:** Es cada una de las instrucciones que escribimos en una interfaz CLI para comunicarnos.
- **Parámetro:** Son las diferentes opciones que puede tomar un comando para alterar su funcionamiento.
- **Script:** Es un grupo de comandos, creado con la intención de automatizar tareas.

Interfaz de usuario

Una **interfaz de usuario** (UI) es como se denomina al medio que utiliza un usuario para comunicarse con una máquina. En nuestro caso nos vamos a referir a **interfaces de línea de comandos** (CLI) cuando son basadas en texto, donde los usuarios escriben órdenes o comandos y el sistema las lee y realiza, o devuelve resultados e información.

Por otra parte, **las interfaces gráficas de usuario** (GUI) son la evolución natural de éstas, utilizando pantallas gráficas o sistemas táctiles, mucho más sencillos y agradables de utilizar para la mayoría de los usuarios, sin embargo, habitualmente menos apropiadas y flexibles en contextos informáticos de desarrollo o sistemas.

Intérprete de comandos

Como mencionamos en un apartado anterior, el intérprete de comandos (también llamado **shell**) es el software o programa que gestiona los comandos que introduce el usuario, para intentar comprenderlos lo mejor posible y comunicárselos a la máquina o sistema de la mejor forma posible. Por ejemplo, una interfaz de línea de comandos puede utilizar varios intérpretes diferentes, como por ejemplo: **Bash o Zsh**.

En GNU/Linux, existen multitud de intérpretes de comandos, siendo quizás los más conocidos bash y zsh. Sin embargo, existen multitud de ellos, como los que puedes ver a continuación:

- **Bourne Shell: sh** Intérprete ligero y sencillo, considerado el estándar.
- **C Shell: csh** Intérprete basado en el lenguaje de programación C.
- **Korn Shell: ksh** Intérprete de AT&T de Bourne Shell.
- **Almquist Shell: ash** Intérprete ligero y rápido, popularizado en máquinas de bajo coste.
- **Bourne Again Shell: bash** Popular intérprete de la mayoría de los GNU/Linux.
- **Z Shell: zsh** Intérprete shell mejorado, con añadidos de otras shells.
- **Fish Shell: fish** Intérprete enfocado en la sencillez y una interfaz agradable.
- **Ion Shell: ion** Shell del sistema operativo Redox.

Terminal vs emulador de terminal

Es también muy frecuente utilizar la palabra terminal en lugar de emulador de terminal. Se hace para acortar, porque en realidad una terminal es un aparato que antiguamente se utilizaba para comunicarse con otro sistema, mientras que un emulador de terminal, que es lo que realmente utilizamos en nuestros dispositivos, es un software que simula ser una de esas terminales, habitualmente con apariencia de ventana de línea de comandos, aunque se puede encontrar de todo, como EDEX-UI.

¿Qué es un comando?

Solemos utilizar la palabra comando para referirnos a la instrucción que escribimos en el emulador de terminal para comunicarnos con el sistema, mientras que opciones o parámetros son los detalles que se suelen escribir a continuación para personalizar la forma de trabajar del comando o alterar su funcionamiento.

En el día a día del uso de terminal utilizamos múltiples comandos, y es conveniente aprender a utilizarlos o al menos los más frecuentes.

- **pwd** Directorio actual - (print working directory).
- **whoami** Muestra el nombre del usuario actual.
- **help** Ayuda.
- **-h** Ayuda.
- **--help** Ayuda.
- **clear** Limpiar la terminal.
- **ls** Lista los archivos del directorio actual.
- **cd** Cambiarse de directorio.
- **~** Representa el directorio home.
- **/** Representa el directorio raíz.
- **.** Representa el directorio actual.
- **..** Representa el directorio padre, subir un nivel del actual.
- **-** Representa regresar al directorio anterior.
- **""** Directorios o archivos con espacios en blanco deben nombrarse entre comillas.
- **touch** Crea un archivo.
- **echo** Crea un archivo con contenido.
- **cat** Visualiza el contenido de un archivo en la terminal
- **mkdir** Crea un nuevo directorio.
- **rm** Elimina un archivo.
- **rmdir** Elimina un directorio vacío.
- **rm -r** Elimina un directorio y su contenido.
- **rm -rf** Forza a eliminar un directorio y su contenido.
- **mv** Mueve archivos o directorios.
- **cp** Copia archivos.
- **cp -r** Copia directorios y su contenido.
- **find** Busca archivos o directorios.
- **ps** Lista procesos activos.
- **kill** Elimina un proceso activo.
- **vim** Abre el editor VIM.
- **nano** Abre el editor Nano.
- **code** Abre el editor VS Code.
- **alias** Crear un atajo personalizado a un comando largo.
- **unalias** Elimina alias.

¿Qué es un script?

Por último, también es muy común crear scripts, que no son más que ficheros de texto donde se escriben una colección de comandos para realizar una tarea concreta, con la intención de automatizarla y sea mucho más fácil para nosotros realizarla, ya que en lugar de escribir todos esos comandos por separado, sólo tenemos que llamar a ese script.