

## 03. Terminal de Comandos

1. ¿Qué es una Terminal?
2. Tipos de Terminales Según el Sistema Operativo
3. Tabla de Comandos Básicos de Consola para Windows y Mac
4. Trabajando con la terminal.

### 1. ¿Qué es una Terminal?

---

Una **terminal** es una interfaz de texto que permite a los usuarios comunicarse directamente con el sistema operativo a través de comandos. En lugar de usar una interfaz gráfica, la terminal permite que los usuarios introduzcan instrucciones de texto para realizar diversas tareas, desde mover archivos hasta instalar software o revisar el estado del sistema.

La terminal es importante porque:

1. **Acceso directo al sistema:** Proporciona acceso directo a funciones avanzadas del sistema operativo que no siempre están disponibles en la interfaz gráfica.
2. **Automatización y scripts:** Permite crear y ejecutar scripts para automatizar tareas repetitivas, lo cual es especialmente útil en administración de sistemas y desarrollo.
3. **Flexibilidad y control:** Ofrece un control detallado del sistema, ya que se pueden ejecutar comandos con parámetros específicos y obtener resultados precisos.
4. **Herramienta común para desarrolladores:** La terminal es la base de muchas herramientas de desarrollo y es esencial para programadores y administradores de sistemas.

### 2. Tipos de Terminales Según el Sistema Operativo

---

Cada sistema operativo tiene su propia terminal o consola con comandos que pueden variar en estructura y uso:

1. **Terminal de Unix/Linux**
  - **Bash (Bourne Again Shell):** Una de las terminales más populares en sistemas Unix y Linux. Permite ejecutar comandos Unix y es ampliamente usada en sistemas Linux.
  - **Zsh:** Similar a Bash, con características adicionales y opciones de personalización. Es la terminal predeterminada en macOS desde 2019.
  - **Fish:** Una alternativa moderna que ofrece más facilidad de uso y autocompletado avanzado.
2. **Terminal de macOS**
  - **Terminal de macOS:** Mac utiliza una terminal similar a la de Unix, ya que su sistema está basado en Unix. Hasta hace poco, usaba **Bash** como shell predeterminado, pero ahora utiliza **Zsh**.
3. **Terminal de Windows**
  - **CMD (Command Prompt):** La terminal tradicional de Windows. Acepta comandos específicos de Windows, pero no es compatible con comandos Unix por defecto.
  - **PowerShell:** Una terminal más avanzada que combina comandos de Windows con una estructura de comandos más poderosa y flexible. Compatible con scripts y muy útil para tareas de administración de sistemas.
  - **WSL (Windows Subsystem for Linux):** Permite instalar un entorno Linux en Windows, ejecutando una terminal Bash para usar comandos y herramientas de Linux directamente en Windows.

## Resumen

La terminal es una herramienta clave para interactuar con el sistema operativo de una manera poderosa y flexible. Según el sistema operativo, hay distintos tipos de terminales, cada una con comandos y funciones específicas. Conocer la terminal adecuada y sus comandos es esencial, especialmente para desarrollo de software y administración de sistemas, donde la eficiencia y la precisión son fundamentales.

### 3. Tabla de Comandos Básicos de Consola para Windows y Mac

#### Gestión de archivos y directorios:

Comando	Descripción	PowerShell	Bash (Linux/macOS)	CMD (Windows)
<code>pwd</code>	Muestra la ruta actual	✓	✓	✗ (Usar <code>echo %cd%</code> )
<code>ls</code>	Lista archivos y carpetas	✓ (también <code>Get-ChildItem</code> )	✓	✗ (Usar <code>dir</code> )
<code>find</code>	Busca archivos y carpetas en el sistema.	✓ ( <code>Get-ChildItem -Recurse</code> )	✓	✓ (Solo búsqueda de texto; <code>dir /s</code> para buscar archivos)
<code>mkdir nombre_carpeta</code>	Crea un nuevo directorio	✓	✓	✓
<code>rm</code>	Elimina archivos o carpetas.	✓ ( <code>Remove-Item</code> )	✓	✗ (Usar <code>del</code> para archivos y <code>rmdir /s</code> para carpetas)
<code>rmdir nombre_carpeta</code>	Borra un directorio vacío	✓	✓	✓
<code>cd [nombre_carpeta, ..]</code>	Cambia de directorio ( <code>cd ..</code> para directorio superior)	✓	✓	✓
<code>touch nombre_archivo</code>	Crea un archivo nuevo vacío	✗ (Usar <code>New-Item</code> )	✓	✗ (Usar <code>echo.&gt; nombre_archivo</code> )
<code>rm -rf nombre</code>	Elimina un archivo o carpeta	✓ (también <code>Remove-Item</code> )	✓	✗ (Usar <code>del</code> para archivos y <code>rmdir /s</code> para carpetas)
<code>cp</code>	Copia archivos o carpetas de un lugar a otro, con el formato <code>cp &lt;origen&gt; &lt;destino&gt;.</code>	✗ (Usar <code>Copy-Item</code> )	✓	✗ (Usar <code>copy</code> )
<code>mv</code>	Mueve archivos o carpetas de un lugar a otro, con el formato <code>mv &lt;origen&gt; &lt;destino&gt;.</code>	✓ ( <code>Move-Item</code> )	✓	✗ (Usar <code>move</code> )
<code>nano nombre_archivo</code>	Abre el editor de texto Nano	✗ (No disponible)	✓	✗ (Usar <code>notepad nombre_archivo</code> )

#### Procesos:

Comando	Descripción	PowerShell	Bash (Linux/macOS)	CMD (Windows)
<code>clear</code>	Limpia la consola ( <code>Ctrl + L</code> como acceso rápido)	✓ ( <code>Clear-Host</code> )	✓	✗ (Usar <code>cls</code> )
<code>tasklist</code>	Muestra la lista de tareas en ejecución	✓ ( <code>Get-Process</code> )	✗ (Usar <code>ps</code> en Bash)	✓
<code>control panel</code>	Abre el panel de control	✓ ( <code>Start-ControlPanel</code> )	✗ (No disponible)	✓
<code>history</code>	Muestra el historial de comandos ejecutados	✓ ( <code>Get-History</code> )	✓	✗ (Usar <code>doskey /history</code> )
<code>exit</code>	Cierra la consola	✓	✓	✓

## Sistema:

Comando	Descripción	PowerShell	Bash (Linux/macOS)	CMD (Windows)
<code>help</code>	Muestra todos los comandos disponibles	✓ ( <code>Get-Help</code> )	✓ ( <code>man</code> )	✓
<code>whoami</code>	Muestra el nombre de usuario en sesión	✓	✓	✓
<code>[paquete] --version</code>	Muestra la versión de un paquete	✓	✓	✗
<code>systeminfo</code>	Muestra la información del sistema	✓ ( <code>Get-ComputerInfo</code> )	✗ (Usar <code>uname -a</code> )	✓
<code>driverquery</code>	Muestra todos los drivers instalados	✓	✗ (No disponible)	✓

## 4. Trabajando con la terminal

---

### Consejos antes de empezar

Las terminales de comandos de cada sistema operativo son diferentes. Unos consejos de productividad a la hora de trabajar con la terminal de comandos:

- Utilizar las flechas arriba y abajo para encontrar comandos ya ejecutados
- Con el tabulador puedes autocompletar los nombres
- Ejecutando “Control + l” limpias la consola de Git - En caso de ejecutar algún proceso, podemos pararlo con “Control + c”