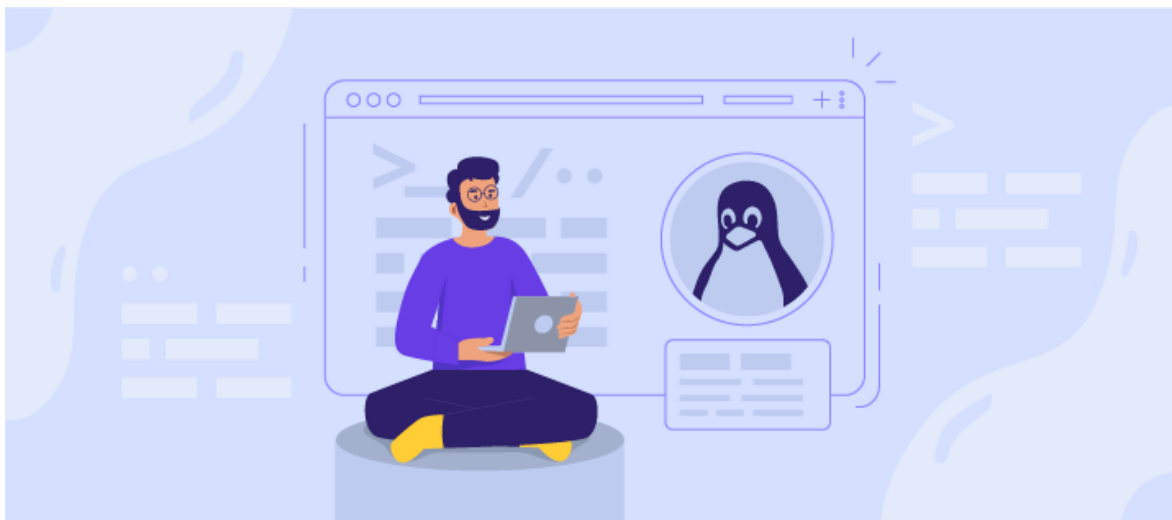




CICLO FORMATIVO DE GRADO SUPERIOR - TÉCNICO EN ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN REDES

IMPLANTACIÓN DE SISTEMAS OPERATIVOS

COMANDOS BÁSICOS EN UBUNTU



1. Qué es una terminal en Linux

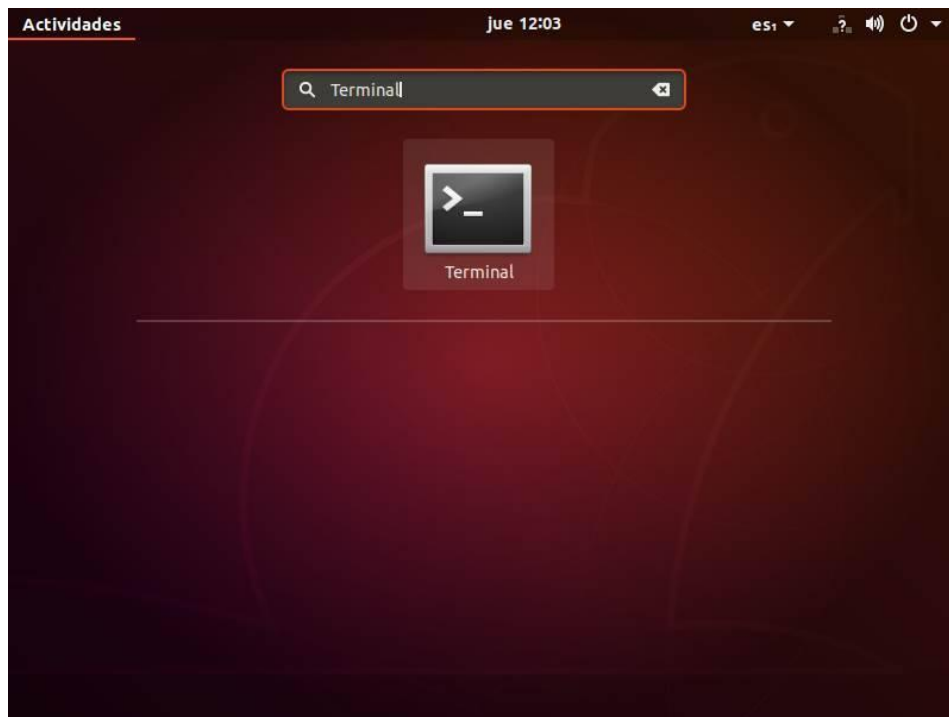
La **Terminal de Linux** es una consola, similar a CMD o PowerShell (pero mucho más avanzada que ambas), utilizada para permitir a los usuarios más avanzados y técnicos controlar hasta el más mínimo detalle del sistema operativo.

Desde esta consola podemos ejecutar todo tipo de binarios o script, especialmente aquellos que no tienen interfaz gráfica y que deben usarse mediante comandos.

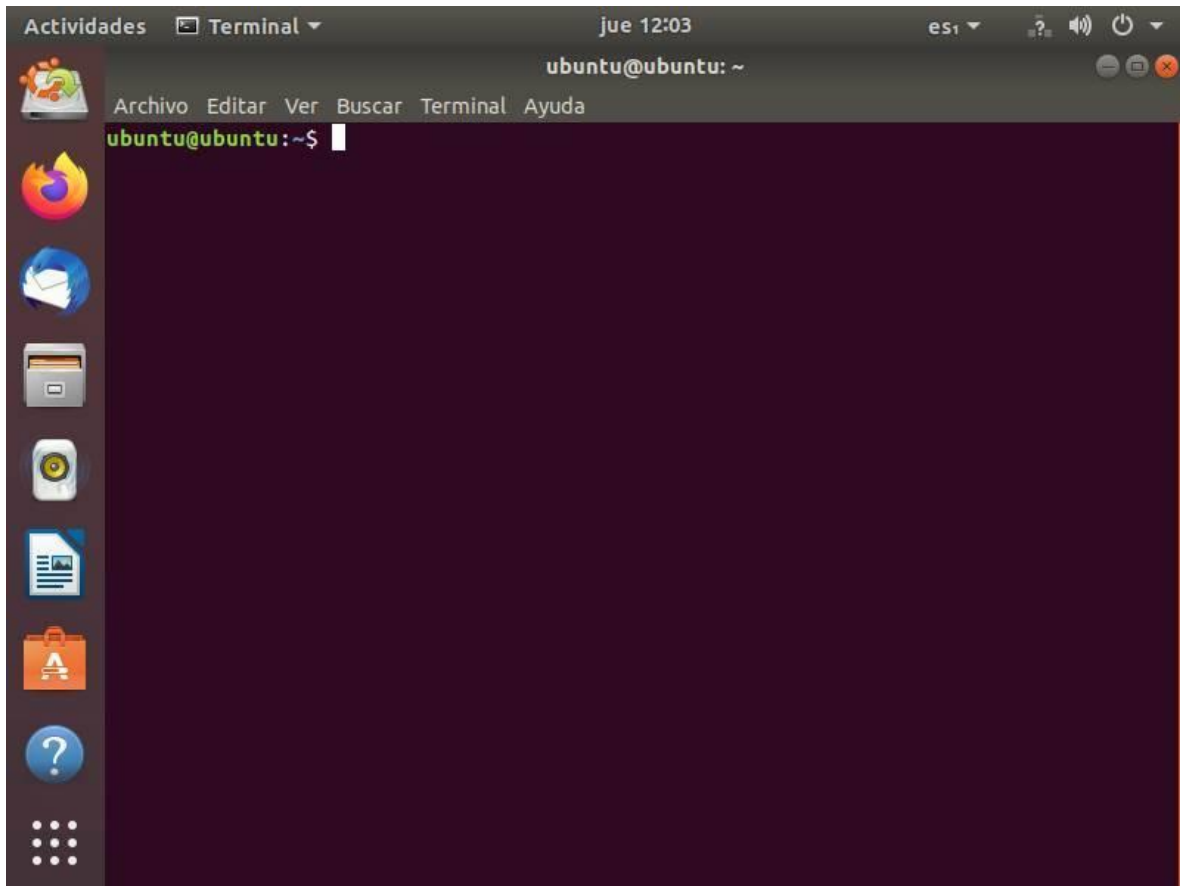
Todo lo que hacemos desde una interfaz, podríamos hacerlo perfectamente desde la terminal, aunque de forma más complicada, larga y menos intuitiva.

2. Abrir una terminal en Linux

Si queremos usar la terminal, podemos encontrar una app dedicada, que se ejecuta en una ventana, dentro del panel de aplicaciones. En el caso de Ubuntu, por ejemplo, podemos encontrar este terminal dentro del cajón de programas del entorno gráfico GNOME.



3. Comandos básicos para Terminal



Cada programa de Linux tiene asociado un nombre, el nombre de su paquete.

Prueba a introducir 'firefox' en el terminal. Verás como se abre directamente el navegador.

DIRECTORIOS

Algo esencial para poder empezar a defendernos con la terminal de Linux es saber cómo movernos por los distintos directorios. Lo primero que nos llamará la atención es el símbolo ~ que aparece a la izquierda de los comandos. Este indica el directorio inicial (/home/usuario) donde hemos empezado.

1. Comando cd

Usando el comando «cd» podemos cambiar de directorio y movernos por todo el árbol de nuestro disco duro. Nos pedirá la ruta completa o el nombre del directorio, dependiendo del directorio de trabajo actual en el que nos encontremos.

Ejemplo. Podemos ir a la carpeta «Descargas» que hay dentro de «/home/usuario/» ejecutando el comando:

```
cd Descargas
```

También podemos usar rutas completas, como:

```
cd /home/usuario/Descargas
```

Hay algunas variaciones:

- **cd ..** (con dos puntos) para ir un directorio hacia arriba.
- **cd** para ir directamente a la carpeta de inicio.
- **cd-** (con un guión) para ir al directorio anterior.

Ojo: el shell de Linux distingue entre mayúsculas y minúsculas. Por lo tanto, debes escribir el nombre del directorio de forma exacta.

Accede a la carpeta Descargas desde el terminal.

2. Comando ls

Una vez que estamos en el directorio que queremos, podemos usar el comando «ls» para ver una lista con el contenido o todos los archivos que hay en ese directorio.

Si escribimos ls, se mostrará el contenido de la carpeta actual. Si queremos ver el contenido e otros escritorios, escribimos ls y luego la ruta del directorio. Ejemplo:
ls/home/pepe/Documentos para ver el contenido de Documentos.

Hay algunas variaciones:

- `ls -R`: también lista todos los archivos en los subdirectorios-
- `ls -a`: muestra los archivos ocultos.
- `ls -al`: lista los archivos y directorios con información detallada como los permisos, el tamaño, el propietario, etc.

Usa el comando `ls` para ver los documentos de la carpeta Descargas a la que hemos accedido desde el terminal.

(Comprueba que haya documentos en dicha carpeta y si no hay, descarga un par de documentos o imágenes de internet antes de ejecutar el comando)

3. Comando `mkdir`

Este comando nos permite crear una nueva carpeta en el directorio en el que estamos ubicados. Debemos ejecutar este comando seguido del nombre de la nueva carpeta que queremos crear, por ejemplo:

`mkdir trimestre1`

Crea una carpeta dentro de la carpeta Documentos y comprueba que se ha creado.

4. Comando `cp`

Con este comando podemos copiar un archivo de un directorio a otro. Podemos usarlo para copiar un archivo del directorio actual a otro directorio, o introduciendo dos rutas absolutas:

Ejemplo: **`cp linux.jpg/home/pepe/Imágenes`** crea una copia de `linux.jpg` en el directorio de `Imágenes`.

5. Comando mv

Este comando funciona igual que el comando cp que acabamos de ver, con la diferencia de que, en vez de copiar el archivo, lo mueve (es decir, desaparece del origen). Este comando también se puede usar para cambiar el nombre a un archivo. Simplemente debemos moverlo al mismo directorio con un nombre diferente. Por ejemplo:

mv archivo.txt /home/pepe/Documentos

Para cambiar el nombre del archivo: mv nombreviejo.ext nombrenuevo.ext

6. Comando cat

Para ejecutar este comando, tecleamos el comando cat seguido del **nombre del archivo** y su extensión correspondiente. Estos son algunos de los ejemplos que podemos usar y que nos ayudarán para hacernos una idea de su nomenclatura y modo de uso:

- cat archivo.txt.
- cat > nombearchivo: para crear un nuevo archivo desde cero.
- cat nombearchivo1 nombearchivo2 > nombearchivo3: une dos archivos (1 y 2) y almacena su salida en el archivo 3.

7. Comando rm

Este nos permite eliminar cualquier archivo usando rutas relativas (es decir, un archivo dentro de nuestra ruta actual, que veamos con ls), o rutas absolutas, moviéndonos hasta el directorio donde se guarda el archivo.

- rm archivonuevo
- rm /home/usuario/Descargas/archivonuevo

También podemos usarlo para eliminar directorios completos:

rm -rf /home/usuario/directorio

Nota: Ten mucho cuidado con el comando y verifica el directorio en el que te encuentras. Este comando elimina todo y no se puede deshacer.

Crea un archivo y guárdalo dentro de la carpeta Documentos. Posteriormente bórralo usando el comando nombrado.

8. Comando pwd

Este comando muestra el directorio en el que estamos trabajando. Permite encontrar la ruta del directorio (carpeta) de trabajo actual en el que nos encontramos). Muy útil cuando nos hemos desplazado (recordamos, con CD) por varios subdirectorios y no nos acordamos exactamente en cuál estamos trabajando. El comando devuelve una ruta absoluta, ejemplo: /home/pepe

9. Comando touch

Permite crear un nuevo archivo en blanco a través de la línea de comando de Linux.

Ingresa `touch /home/nombredeusuario/Documentos/Web.html` para crear un archivo HTML titulado **Web** en el directorio **Documentos**.

10. Comando locate

Permite **localizar** un archivo, al igual que el comando de búsqueda en Windows. Además, el uso del argumento **-i** junto con este comando hará que no distinga entre mayúsculas y minúsculas, por lo que puedes buscar un archivo incluso si no recuerdas su nombre exacto.

Para buscar un archivo que contenga dos o más palabras, usa un asterisco (*). Por ejemplo, el comando **locate -i escuela*nota** buscará cualquier archivo que contenga la palabra «escuela» y «nota», ya sea en mayúsculas o minúsculas.

11. Comando find

find también permite buscar archivos y directorios.

Como ejemplo, el comando **find /home/ -name notas.txt** buscará un archivo llamado **notas.txt** dentro del directorio de inicio y sus subdirectorios.

Otras variaciones al usar **find** son:

- Para buscar archivos en el directorio actual, **find . -name notas.txt**
- Para buscar directorios, **/ -type d -name notes.txt**

12. Comando grep

Permite buscar a través de todo el texto en un archivo dado.

Ejemplo: **grep rojo notepad.txt** buscará la palabra rojo en el archivo del bloc de notas.

Las líneas que contienen la palabra buscada se mostrarán.

13. Comando sudo

Abreviatura de «**SuperUser Do**» (SuperUsuario hace), este comando permite realizar tareas que requieren permisos administrativos o raíz. Sin embargo, no es aconsejable usar este comando para el uso diario, ya que podría ser fácil que ocurra un error si haces algo mal.

14. Comando df

Se usa el comando **df** para obtener un informe sobre el uso del espacio en disco del sistema, que se muestra en porcentaje y KB. Para ver el informe en megabytes, escribe **df -m**.

Ejecuta este comando y observa el resultado en pantalla.

Comandos del sistema

Linux también cuenta con una gran variedad de comandos que nos permiten controlar la mayoría de los aspectos del sistema operativo. Algunos de los más sencillos y que nos merece la pena conocer son:

- **date:** muestra la fecha y la hora.
- **uptime:** muestra el tiempo que lleva encendido el ordenador.
- **uname -a:** nos muestra la versión del kernel.
- **whoami:** nos indica cuál es nuestro usuario actual.
- **chmod:** nos permite cambiar los permisos de cualquier archivo o directorio.

Para conocer los distintos modos de empleo, ejecuta 'chmod --help'

- **chown:** nos permite cambiar la autoría de cualquier archivo o directorio.

Para conocer los distintos modos de empleo, ejecuta 'chmod --help'

- **mount:** nos permite montar unidades (discos duros, USB, tarjetas de memoria, etc) en un directorio.
- **clear:** Limpia la Terminal por completo.

Prueba cada uno de estos comandos en el Terminal y comprueba qué sucede.

Para la gestión de usuarios:

- **su:** permite iniciar sesión como root desde la sesión actual.

O utiliza **sudo -i** si muestra Fallo de autenticación

- **exit:** cierra la sesión de root abierta.
- **adduser:** Añade un nuevo usuario a nuestro Linux.

Usa el comando seguido del nombre del nuevo usuario.

- **passwd:** Permite cambiar la contraseña del usuario que le especifiquemos.

Crea un nuevo usuario y modifica su contraseña.

Para el control:

- **top:** es un visor de procesos que permite ver todo lo que se está ejecutando en el sistema.
- **kill ID:** permite 'matar' cualquier proceso desde su identificador.
- **killall nombreaplicacion.** Con este comando se cierran todos los procesos relacionados con la aplicación la línea de comandos de una forma muy cómoda y rápida.
- **ps:** muestra todos los procesos en ejecución.
- **free:** muestra información sobre la RAM usada y libre.
- **halt:** detiene todos los procesos del sistema y lo apaga. Es recomendable haber guardado todos los archivos en los que estemos trabajando en segundo plano si no queremos perder los cambios.
- **shutdown:** nos permite apagar el PC en un minuto. Podemos usar el parámetro -R para especificar el tiempo, en minutos, para apagar el PC. Y con -H now podemos saltarnos el minuto de espera y apagar el equipo directamente.
- **reboot:** reinicia el ordenador.

Muchos de estos comandos ofrecen las mismas funcionalidades tanto en Windows a través de Command Prompt como en macOS haciendo uso de Terminal.

Abre Firefox e intenta cerrar su proceso desde la terminal con killall seguido del nombre de la aplicación.

Comprueba la información de la RAM desde la terminal.

Reinicia el ordenador con el comando reboot y comprueba que se ha creado el usuario anterior en el S.O.

Instalar y desinstalar programas

Desde la terminal de Linux también se suelen instalar y desinstalar todo tipo de programas. Aquí sí es donde cada distribución de linux cambia, ya que el gestor de paquetes que viene incluido por defecto en Ubuntu no es el mismo que podemos encontrar, por ejemplo, en Arch Linux.

En Ubuntu, el gestor de paquetes que encontramos es APT. Con él, desde Terminal, podemos descargar, instalar, actualizar y eliminar cualquier paquete de software. Las distribuciones vienen con unos repositorios de software bastante completos, por lo que si conocemos el nombre del paquete podemos instalarlo fácilmente de la siguiente manera:

apt install firefox

Para eliminar un programa instalado, podemos cambiar «install» por «remove» (para eliminar solo los paquetes) o «purge» (para eliminar el paquete y todas las configuraciones y demás archivos residuales):

apt purge firefox

También debemos saber cómo añadir nuevos repositorios a Ubuntu y cómo actualizarlos para tener siempre las últimas versiones del software.

Por último, desde la consola también podemos descargar e instalar las últimas actualizaciones para nuestra distro Linux. Siguiendo el ejemplo de Ubuntu, podemos descargar las últimas actualizaciones para nuestra versión de Ubuntu (tanto del sistema como de las aplicaciones instaladas) con:

- **sudo apt-get update**
- **sudo apt-get upgrade**

Y si hay una nueva versión de Ubuntu podemos descargarla e instalarla con los siguientes comandos:

- **sudo apt-get update**

- **sudo apt do-release-upgrade**

Algunas distribuciones pueden usar otros gestores de paquetes, como pacman, o hacernos recurrir directamente a los instaladores como «dpkg», «yum» y «rpm».

Comandos de red

También podemos usar la Terminal de Linux para realizar consultas y peticiones a través de Internet. Los comandos más conocidos, utilizados y útiles que podemos encontrar dentro de esta categoría son:

- **ifconfig**: nos permite ver una lista con todas las direcciones IP de nuestro PC.
- **ping** [servidor o IP]: envía un ping a un servidor (Google, en nuestro ejemplo), comprueba si hay respuesta y mide la latencia.
- **traceroute**: traza la ruta completa desde nuestro PC hasta el servidor.
- **whois** [servidor]: obtiene información acerca de un dominio (en nuestro caso, Google).
- **wget** [url]: nos permite descargar el archivo que le especifiquemos en la URL.
- **ftp** [servidor o IP]: nos permite conectarnos a un servidor FTP.
- **iptables**: nos permite abrir el administrador del firewall nativo de Linux.

4. Programas de terminal

En la red existen una gran cantidad de programas de terminal, cada uno con unas características únicas, de manera que cada usuario pueda elegir la consola que más le guste o mejor se adapte a sus necesidades o forma de trabajar, pero, al final, independientemente de cuál utilicemos, vamos a poder realizar las mismas funciones a través de la línea de comandos, ya que lo único que cambia es la estética, no la funcionalidad que nos pueda ofrecer una u otra versión de Linux.

Algunos ejemplos de programas son los siguientes:

- **GNOME Terminal:** La más conocida y utilizada. Esta consola forma parte de las herramientas que vienen incluidas en el escritorio GNOME de Linux, escritorio que viene por defecto en muchas distros, como, por ejemplo, Ubuntu.

sudo apt-get install gnome-terminal

- **XTerm:** Este terminal viene instalado junto al servidor de ventanas X Window, y ofrece un TTY sencillo.

sudo apt-get install xterm

- **Konsole:** Cuenta con una gran cantidad de opciones de personalización, y también tiene soporte para pestañas y marcadores, lo que facilita el trabajo a los usuarios.

sudo apt-get install konsole

Existen otros como Terminator, Alacritty, Lilyterm, Tilda, etc

Elige el programa de terminal que más te gusta e instálalo.

5. ¿Qué son los Huevos de Pascua de terminal?

Son funciones o trucos ocultos dentro de un programa de manera que este solo tenga lugar si se realiza una tarea concreta. Los huevos de pascua son bastante comunes en los programas y en el propio sistema operativo Linux, Easter Eggs que podemos lanzar fácilmente desde un terminal.

La vaca y el toro son dos de los animales favoritos de los desarrolladores de Linux, y tienen sus propios huevos de Pascua.

¿Cómo lanzamos el toro? ejecutamos: **apt-get moo**.

Ejecuta y observa lo que aparece en pantalla.

¿Cómo lanzamos la vaca? Ejecutamos **sudo apt-get install cowsay**.

Instala cowsay ejecutando el comando anterior. Una vez esté instalado escribe en la terminal cowsay "Aquí estoy pastando"

Curiosidad. Hay un huevo de pascua que permite escribir un insulto por parte de Linux cuando un usuario introduce mal la contraseña de «sudo». Para ello, ejecutamos el comando **sudo visudo**, escribiremos arriba del todo «Defaults insults», sin comillas, guardamos y salimos.

Busca otros huevos de pascua y pruébalos.