

Módulo Herramientas de Desarrollo

La terminal de Linux

Objetivos: Entender cómo manejarse en la terminal de Linux y aprender sus comandos básicos.

La línea de comandos es donde pasaremos la mayor parte del tiempo al conectarse a otros servidores y servicios en la nube. Gracias a ésta, se pueden ejecutar comandos complejos sin necesidad de una interfaz gráfica. Es importante saber usar la terminal ya que nos permite realizar acciones que serían muy complejas o de plano imposibles de hacer de forma gráfica. También podemos automatizar tareas y organizar recursos que de otra forma tomarían mucho tiempo realizar.

No es necesario saber de memoria todos los comandos, pero es importante saber en general cómo moverse en la consola y cómo obtener información de comandos que necesitemos. En el desarrollo de software es común el uso de comandos para instalar dependencias, construir aplicaciones y realizar tests, entre otras actividades.

Algo importante a tener en cuenta antes de empezar: A diferencia de Windows, **Linux es sensible a mayúsculas y minúsculas**. Si un comando se escribe en minúsculas, NO funcionará si usamos mayúsculas.

Para abrir la terminal, podemos ir al listado de aplicaciones y hacer clic en el icono de la aplicación:




Alternativamente, simplemente presionando **CTRL+ALT+T** se abrirá una ventana nueva de la terminal.

Copiar y pegar en la consola

Los atajos de teclado en la consola para copiar y pegar son ligeramente diferentes a lo que estamos acostumbrados. **CTRL+SHIFT+C** sirve para copiar y **CTRL+SHIFT+V** para pegar. También podemos hacer uso del botón central del mouse para pegar.



Autocompletar



La tecla **Tab**  sirve para autocompletar. Esto nos permite tipear menos para conseguir lo que queremos.

Por ejemplo, si queremos ir a un directorio "~/Documentos/notas/ubuntu", podemos comenzar a tipear "Do" y con solo pulsar la tecla "Tab" la consola sabrá que queremos decir "Documentos". Si hay otra carpeta que cumple los requisitos, nos dará una lista a elegir y usar tab nos permitirá movernos por la lista hasta llegar a la que queremos. Entonces, para llegar a esa carpeta, solo tendremos que tipear: "Do", Tab, "n", Tab, "u" Tab. Si solo hay una opción posible, incluso podemos hacerlo más rápido, simplemente pulsando Tab nos llenará automáticamente el nombre de la carpeta. La función de autocompletar también ayudará al usuario a la hora de ejecutar comandos con argumentos y opciones.

Uso de las teclas de dirección

Las teclas de dirección son extremadamente útiles a la hora de manejarse en la consola. El mouse tiene poco protagonismo, por lo que tenemos que valernos más del teclado para ejecutar acciones.

Las flechas horizontales   sirven para movernos un carácter a la vez en la instrucción que estamos tipeando. Si usamos **CTRL+←** o **CTRL+→** nos moveremos palabra por palabra horizontalmente. También podemos usar las teclas **HOME/INICIO** y **END/FIN** para movernos al comienzo y al final de una

Por otro lado, las flechas verticales   sirven para navegar instrucciones que ejecutamos anteriormente, esto es particularmente útil cuando debemos ejecutar un mismo comando en repetidas ocasiones.

Navegación

Existen varios comandos que nos ayudan a ubicarnos en un directorio en particular y obtener información del mismo.

El comando ls

Comenzamos con **ls**, que listará todos los archivos y directorios en el directorio actual.

```
ls
```

```
Descargas  Escritorio  Música      Público  Vídeos
Documentos  Imágenes    Plantillas  snap
```

Este comando admite varias opciones, las más comunes son **ls -l**, que mostrará los archivos con el formato largo:

```
ls -l
```

```
total 36
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Descargas
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Documentos
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Escritorio
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Imágenes
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Música
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Plantillas
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Público
drwx----- 3 xacademy xacademy 4096 dic  7 14:10 snap
drwxr-xr-x 2 xacademy xacademy 4096 dic  7 14:15 Vídeos
```

Otras opciones comunes son “**-a**” que muestra los archivos ocultos (cuyo nombre comienza con un punto ‘.’, como por ejemplo un archivo de ambiente .env) y “**-h**” que muestra los tamaños de archivo de una manera más legible.

Finalmente, se le puede pasar una ruta para listar los archivos y directorios dentro de esa ruta, como por ejemplo **ls /** para listar el contenido del directorio raíz.

```
ls /
```

```
bin    dev    lib    libx32  mnt    root   snap    sys    var
boot   etc    lib32  lost+found  opt    run    srv      tmp
cdrom  home  lib64  media   proc   sbin   swapfile  usr
```

El comando pwd

El siguiente comando a tener en cuenta es **pwd** (“print working directory” en inglés) que muestra el directorio en el cual estamos posicionados actualmente. Nos permite orientarnos en el sistema de archivos y asegurarnos de que estamos donde creemos estar.

```
pwd
/home/xacademy
```

El directorio por defecto será el de nuestro directorio de inicio, que se ubica en `/home/{nuestro nombre de usuario}`.

El comando `cd`

Si queremos movernos a otro directorio, hacemos uso del comando **cd** ("change directory"). Por ejemplo, digamos que queremos movernos al directorio raíz y luego ejecutamos `pwd` para verificar que estemos realmente en ese directorio.

```
cd /
pwd
/
```

Si queremos regresar a nuestro directorio de inicio, podemos hacerlo usando **cd** sin argumentos

```
cd
pwd
/home/xacademy
```

También podemos movernos un directorio arriba usando `cd ..`

```
cd ..
pwd
/home
```

Manipulación

Existen varios comandos que nos sirven para modificar los archivos.

El comando `mkdir`

El comando `mkdir` se usa para crear directorios. Por ejemplo:

```
cd
mkdir nueva_carpeta
ls
```

Descargas	Escritorio	Música	Plantillas	snap
Documentos	Imágenes	nueva_carpeta	Público	Videos

El comando touch

El comando touch se usa para crear archivos vacíos. Por ejemplo:

```
touch archivo_vacio.md
ls
archivo_vacio.md  Documentos  Imágenes  nueva_carpeta  Público  Videos
Descargas        Escritorio  Música    Plantillas    snap
```

Los comandos cp y mv

Los comandos cp y mv funcionan muy parecidos, uno copia y el otro mueve. El comando mv también puede ser usado para renombrar un archivo o directorio.

Primero veamos un ejemplo con cp, donde copiamos el archivo vacío que creamos con touch a la carpeta “nueva_carpeta” que creamos con mkdir. Es necesario colocar el nombre del archivo de destino:

```
cp archivo_vacio.md nueva_carpeta/copia_archivo.md
ls
archivo_vacio.md  Documentos  Imágenes  nueva_carpeta  Público  Videos
Descargas        Escritorio  Música    Plantillas    snap
ls nueva_carpeta
copia_archivo.md
```

El comando “mv” funciona igual, toma un archivo de origen y uno de destino, con la diferencia que mueve el archivo en lugar de copiarlo. Podemos comprobarlo listando los archivos con ls luego de usar mv:

```
mv archivo_vacio.md nueva_carpeta/archivo_movido.md
ls
Descargas  Escritorio  Música    Plantillas  snap
Documentos Imágenes   nueva_carpeta  Público    Videos
ls nueva_carpeta
archivo_movido.md  copia_archivo.md
```


El comando rm

Para eliminar archivos o directorios usamos el comando rm. Para eliminar archivos basta con usar rm y la ruta del archivo a eliminar.

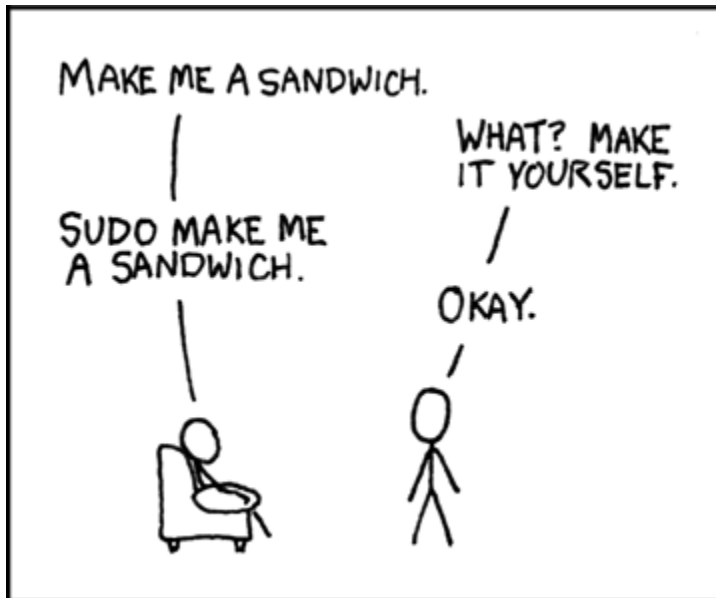
```
touch archivo_a_eliminar.md
ls
archivo_a_eliminar.md  Documentos  Imágenes  nueva_carpeta  Público  Vídeos
Descargas              Escritorio  Música     Plantillas    snap
rm archivo_a_eliminar.md
ls
Descargas  Escritorio  Música      Plantillas  snap
Documentos Imágenes   nueva_carpeta Público     Vídeos
```

Sin embargo, si deseamos eliminar un directorio, es necesario hacer uso de la opción “-r” para que remueva de forma recursiva todos los archivos y directorios dentro de ese directorio.

```
rm nueva_carpeta/
rm: cannot remove 'nueva_carpeta/': Is a directory
rm -r nueva_carpeta/
ls
Descargas  Escritorio  Música      Público  Vídeos
Documentos Imágenes   Plantillas  snap
```

Uso de “sudo”

Al agregar “sudo” antes de un comando, le permite a un usuario ejecutar ese comando como el súper usuario. Existen comandos y archivos que para ser accedidos o ejecutados, requieren permisos especiales y sin estos, el sistema operativo no nos permitirá accederlos.



Por ejemplo, un comando útil es “cat”, el cual nos muestra los contenidos de un archivo. Digamos que queremos ver el listado de contraseñas encriptadas de usuarios de sistema, que se guarda en el archivo `/etc/shadow`.

```
cat /etc/shadow
/var/log/syslog: Permission denied
```

Como este archivo tiene permisos solo de lectura para el usuario root, necesitamos sudo para siquiera leerlo.

```
sudo cat /etc/shadow
[sudo] password for xacademy:
```

Al usar sudo, nos pedirá la clave del usuario (que al tipearla, no mostrará ni asteriscos, ni nada) y si la ingresamos correctamente, accederemos al archivo en cuestión.

Es muy importante saber lo que se está haciendo cuando se usa sudo, ya que su uso inadecuado puede causar riesgos de seguridad o errores de sistema.

Instalar paquetes usando la terminal

Para agregar funcionalidades a nuestro sistema operativo desde la terminal podemos hacer uso del comando apt, el cual provee una interfaz para el manejo de paquetes de sistema. Por defecto, debemos hacer uso de **sudo** para poder ejecutar apt ya que requiere permisos elevados.

El primer comando que ejecutaremos es **sudo apt update**, esto actualizará el listado de paquetes y versiones disponibles para ser instaladas. NO actualizará los paquetes ya instalados en el sistema (para eso, podemos usar el comando **sudo apt upgrade**).

sudo apt update

```
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [656 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [577 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [781 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4732 B]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [963 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [624 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [8150 B]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [956 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [7278 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3520 B]
```

Luego de actualizar el listado, podremos instalar un paquete, en este caso instalaremos **curl** que es un comando útil para el desarrollo de software en general.

sudo apt install curl

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 49 not upgraded.
Need to get 194 kB of archives.
After this operation, 453 kB of additional disk space will be used.
Get:1 http://ar.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64
7.81.0-1ubuntu1.6 [194 kB]
Fetched 194 kB in 1s (247 kB/s)
Selecting previously unselected package curl.
(Reading database ... 212828 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.6_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.6) ...
Setting up curl (7.81.0-1ubuntu1.6) ...
```


Processing triggers for man-db (2.10.2-1) ...

Obtener documentación de un comando

La herramienta más completa para obtener documentación de un comando en particular es el comando “man”. Nos sirve para leer el manual del comando que queremos usar y trae información útil sobre lo que queramos hacer. Por ejemplo `man ls` nos mostrará el manual del comando “ls”, con información útil sobre las opciones disponibles para usarlo. Para salir del manual solo hace falta pulsar la tecla “q”.

Otra opción es usar el sitio cheat.sh. Podemos usarlo directamente desde la línea de comandos ejecutando `curl cheat.sh/comando`. Por ejemplo:

```
curl cheat.sh/ls
# To display everything in <dir>, excluding hidden files:
ls <dir>

# To display everything in <dir>, including hidden files:
ls -a <dir>

# To display all files, along with the size (with unit suffixes) and timestamp:
ls -lh <dir>

# To display files, sorted by size:
ls -S <dir>

# To display directories only:
ls -d */ <dir>
...
```

Podemos probar usar man y cheat.sh con otros comandos útiles, como “cp”, “mv”, “rm”, “mkdir”, “touch” y “top”. Es importante tener siempre en cuenta este comando, ya que dependiendo de la versión del comando o sistema operativo, puede que el uso de algunos comandos varíen.

