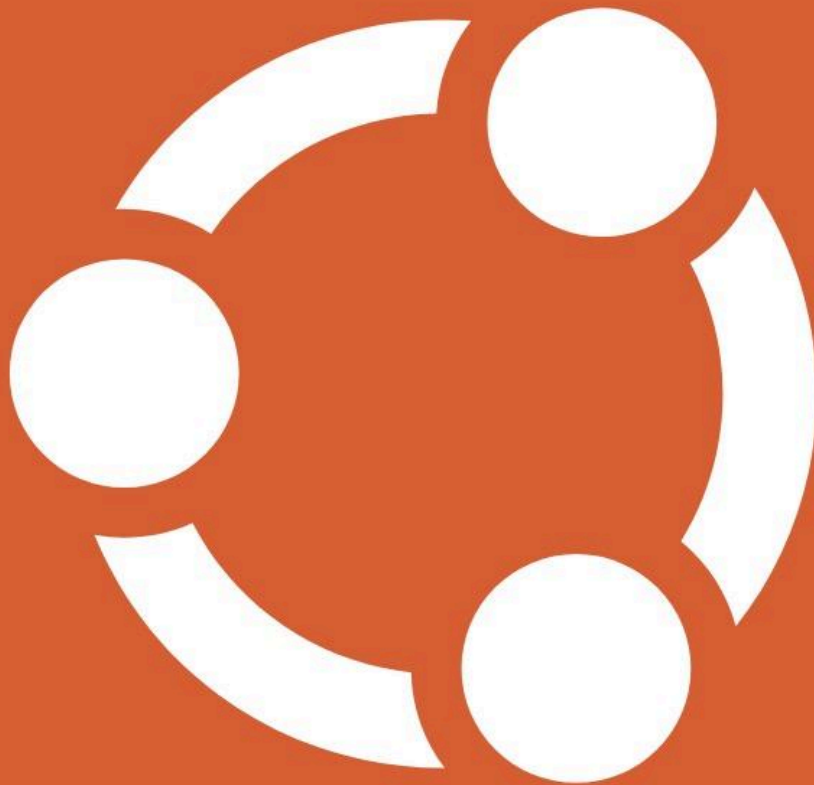


Unidad 7

Ubuntu





ÍNDICE

Objetivo	3
Introducción a Ubuntu	3
Entorno de escritorio GNOME en Ubuntu	4
Sistema de archivos	5
Unidades de disco	5
inodo	5
Directorios principales de Linux	6
Ficheros contenidos en la raíz	7
Ficheros importantes del sistema	7
Ficheros del Kernel	8
Bash	9
Estructura del prompt	9
Sintaxis de un comando	10
Comandos básicos fundamentales	11
🌐 Links comandos 🌐	12
Comodines (Wildcards)	13
Rutas absolutas vs. rutas relativas	13
Standard output (stdout) y standard error (stderr)	14
Enlace duros y enlace simbólicos	15
Sudo	17
Permisos	17
Grupos	20
Tubería	22
Operadores con Tubería	22
TTY y terminales virtuales	23
Instalación de paquetes/aplicaciones	23
Networking	24
Scripting	26
WSL	27
Otros	29
Instalar guest additions en Virtual Box	29
📖 Libro Linux	29
🎮 Tutoriales y juegos aprendizaje Bash 🎮	29
Glosario de términos	30



Objetivo

- Aprender en el funcionamiento de Ubuntu a nivel usuario
- Adquirir habilidades para la administración y configuración avanzada del sistema.
- Aprender a automatizar tareas mediante la línea de comandos y scripts.
- Conocer la configuración de servicios en un servidor Ubuntu.
- Aprender el funcionamiento del sistema en red
- Aplicar principios de seguridad para proteger el sistema y la red.

Introducción a Ubuntu

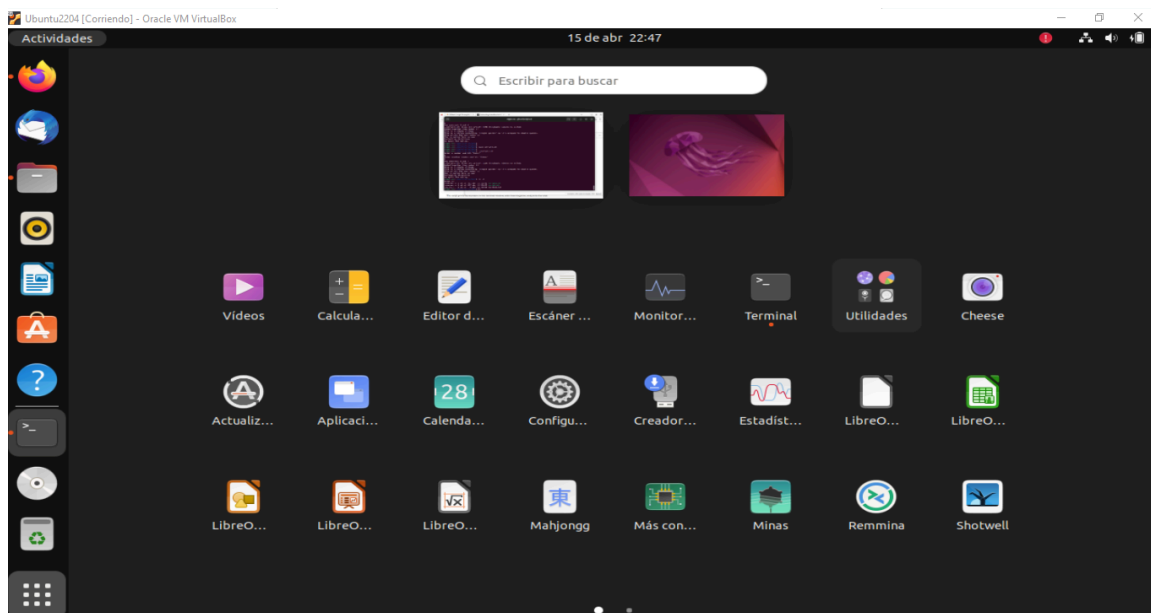
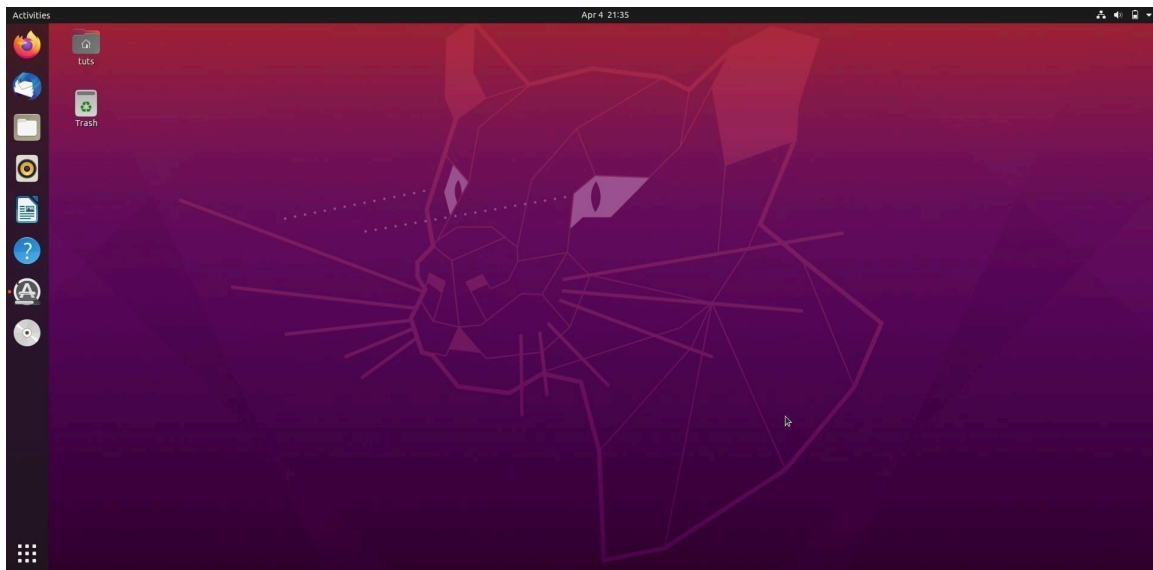
Ubuntu es una distribución GNU/Linux basada en Debian.

- Dispone de versiones de
 - Escritorio
 - Servidor
 - IoT
 - Cloud
 - AWS
 - GCP (Google Cloud)
 - Azure
- La versión estable LTS (Long Term Support) tiene soporte de 5 a 12 años según la versión. Son más estables y serán actualizadas durante ese tiempo.
- La versión PRO incluye mayor soporte de paquetes (gratis hasta 5 máquinas)
- La versión que es de pago para uso profesional se paga por el soporte.
- Soporta múltiples escritorios GUI como GNOME, KDE Plasma, Xfce y LXDE, siendo GNOME la predeterminada (versión customizada por Canonical)
- Opciones de instalación: se puede instalar Ubuntu desde un CD/DVD, una memoria USB o una imagen ISO. Además puede probarse sin instalación.



Entorno de escritorio GNOME en Ubuntu

- Escritorio GNOME
 - Barra de tareas
- Menú de aplicaciones
- Panel de control
 - Configuración
- Notificaciones
- [Nautilus](#) es el administrador de archivos oficial por defecto





Sistema de archivos

Los sistemas tipo Unix no dividen las unidades como en Windows en unidades C:\. En su lugar, tienen un único sistema de archivos unificado, y las unidades individuales se pueden conectar ("montar") en cualquier ubicación del sistema de archivos que tenga más sentido.

El directorio **/**, a menudo denominado directorio raíz (*root directory* en inglés. ⚠ No confundir con usuario root), es la base de este sistema de archivos unificado. A partir de ahí, todo lo demás se ramifica para formar un árbol de directorios y subdirectorios.

 [Fuente](#)

Unidades de disco

Un único directorio raíz representado con **/**

Nomenclatura de los discos duros

- Discos IDE: hd
- Discos SATA: sd

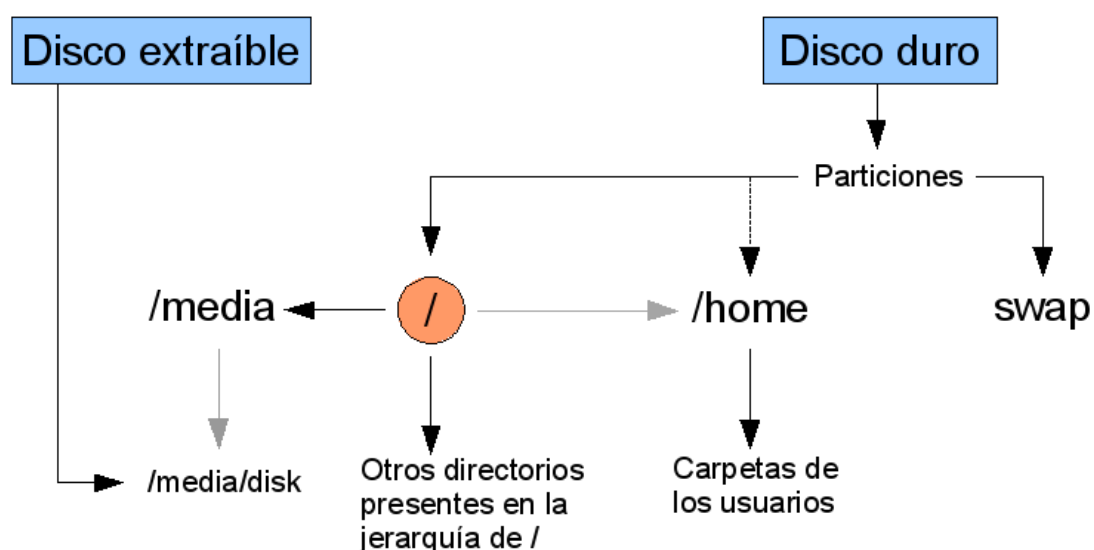
Dependiendo del lugar donde estén conectados (a, b, c...)

Particiones: se indican con un número

- 1...4: reservado para particiones primarias
- 5 en adelante: particiones lógicas

 **File Hierarchy Standard (FHS)** en [wikipedia](#)

Ejemplo de una jerarquía del núcleo Linux





inodo

Es una estructura de datos que contiene la información sobre un archivo o directorio, como permisos, propietario, grupo, tamaño, fechas de acceso/modificación, y punteros a los bloques de datos que contienen el contenido del archivo.

Es como la tarjeta de identificación de un archivo. En lugar de almacenar el nombre del archivo en sí, el inodo guarda información crítica sobre ese archivo. Almacenan archivos regulares, directorios, y enlaces simbólicos. Es identificado por un número entero.

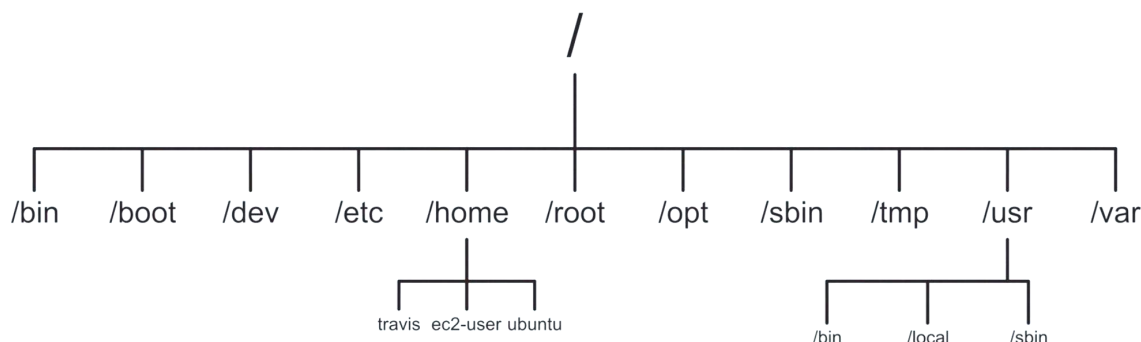
El sistema de archivos Linux mantiene una tabla de inodos que registra la información de todos los archivos y directorios. Cuando se accede a un archivo, el sistema operativo utiliza el número de inodo para localizar la información correspondiente en la tabla de inodo.



Directorios principales de Linux

- /** : Carpeta raíz del sistema
- /bin** : comandos básicos del sistema
- /sbin** : comandos de administración del sistema
- /boot** : ficheros del Kernel y arranque (GRUB)
- /dev** : ficheros de dispositivos (descriptores)
- /etc** : ficheros de configuración del sistema
- /home** : almacena carpetas de usuarios
- /root** : directorio del usuario root
- /opt** : almacena paquetes de aplicaciones que no vienen por defecto
- /tmp** : info. Temporal de la sesión de usuario
- /usr** : programas (aplicaciones), paquetes instalados (**U**nix **S**ystem **R**esources)
- /var** : ficheros de tamaño variable (logs)

⚠ DIFERENCIA entre **/bin** y **usr/bin**



 travis; ec2-user; ubuntu son nombres de usuario ejemplo

- /lib**: librerías usadas por /bin y /sbin
- /mnt**: montaje de dispositivos externos
- /proc**: muestra información del Kernel (FS virt.)
- /lost+found**: para la utilidad fsck (Recovery)
- /media**: montar sistemas de ficheros temporal
- /srv**: ficheros con información del sistema
- /sys**: información de dispositivos (Kernel). Evolución de /proc



Ficheros contenidos en la raíz

initrd.img: para montar el sistema de ficheros

initrd.img.old: copia de backup de initrd.img

swapfile: archivo SWAP (memoria virtual)

vmlinuz: enlace simbólico al Kernel

Directorios especiales en Linux

. → Directorio actual

.. → Directorio padre



Directorios y ficheros ocultos en Linux empiezan por un punto .



Los nombres de archivo y directorio son **CASE SENSITIVE**
(ie., diferencian entre mayúsculas y minúsculas)

Ficheros importantes del sistema

/etc/hostname

/etc/hosts

/etc/passwd

/etc/resolvconf

/etc/crontab

/etc/fstab

/etc/adduser.conf

/etc/apt/sources.list

/etc/shadow

/etc/network/interfaces

.bashrc

.profile

.bash_logout

.bash_history



Ficheros del Kernel

Se ubican en el directorio **/boot**


config	Fichero de configuración del Kernel
initrd.img	RamDisk inicial
System.map	Tabla de símbolos del Kernel
vmlinuz	Fichero del Kernel (está comprimido)
GRUB	Cargador de arranque
https://www.guia-ubuntu.com/index.php/GRUB	
Ficheros memtest	Usados para test de RAM



Bash

Intérprete de línea de comandos que se ejecuta en una ventana de texto donde el usuario escribe órdenes en modo texto actuando así como interfaz con el ordenador.

Se le denomina también shell, terminal (nombre de la aplicación en Ubuntu), consola, prompt, command line entre otros nombres.

 <https://ubuntu.com/tutorials/command-line-for-beginners#1-overview>
 <https://terminaldelinux.com/terminal/>

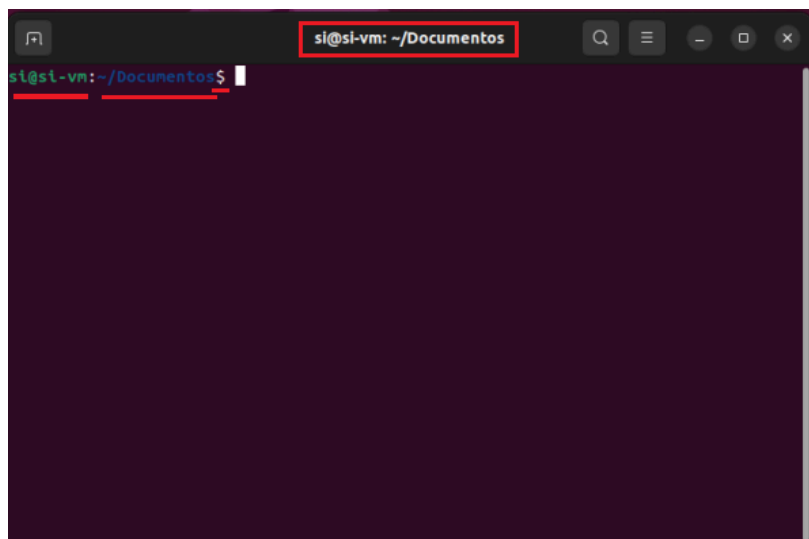
 Se puede leer y ejecutar órdenes desde un archivo, llamado guión o 'script'



Atajo de teclado para abrirla en Ubuntu
Y mayoría de distros

 ctrl + alt + t 

Estructura del prompt



si → nombre usuario
si-vm → nombre equipo
~/Documentos → directorio
~ → Indicador que estás dentro del directorio /home
\$ → Indicador de tipo de usuario
 \$: normal
 #: root



Sintaxis de un comando

`nombre_comando [opciones] [valor]`

Opciones también se denominan **flags** o **argumentos**


Las opciones puede ser:

- Un sólo carácter precedido por un guión. Ejemplo: -u
- Varios caracteres/palabra precedida por doble guiónEjemplo: - - all

Algunos comando tienen versión abreviada -all == a



Los comandos son **CASE SENSITIVE** (diferencia entre mayúsculas y minúsculas).
Pudiendo no funcionar si se cambia, o ser un comando diferente

 Existen muchos comandos **silent**, que no muestran salida de texto tras ejecutarse



Comandos básicos fundamentales

pwd	muestra el directorio en que te encuentras
ls	ls -l (ver permisos directorio) ls -l nombreArchivo ls -a
cd	cd nombreDirectorio cd /home cd .. cd /
whoami	usuario
mkdir nombreDirectorio mkdir /ruta/nombreDirectorio	make directory
mkdir ruta/nombreDirectorio	
help <i>bashcommand</i> <i>comando - -help</i>	
man <i>comando</i>	Muestra la info en una pantalla temporal
whatis <i>comando</i>	
more	muestra contenido por paginas
less	muestra contenido por paginas. permite usar flechas
touch	crear archivo si se precede por punto . será oculto
cat	muestra el contenido de archivos y tb concatena
df -h	disco libre
du -h	
echo	escribir en pantalla
top	muestra procesos de linux
>	
>>	
cp	copia -R (Copia recursiva)
mv	mueve archivos y/o cambia de nombre
rm rmdir	Borra. 🗑️ No hay papelera!



grep	permite buscar palabras o cadenas de texto en archivos o en la entrada estándar de la terminal.
head	mostrar el principio de un archivo
tail	
file	tipo de archivo
!!	retype comando?
kill	acaba con proceso
find	busca tb en subdirectorios
alias	crea alias comando alias foo='ls -l'
ps	procesos
top	procesos

Links comandos

<https://www.linuxtrainingacademy.com/linux-commands-cheat-sheet/>

<https://www.geeksforgeeks.org/linux-commands-cheat-sheet/>



Si separas dos comandos por punto y coma “;”
ambos comandos serán ejecutados (por ese orden).
Incluso se pueden añadir más de 2.



Comodines (Wildcards)

Facilitan el trabajo de administración, sustituyendo uno o varios datos alfanuméricos

- asterisco ***** : cero, uno o varios caracteres
- interrogación **?** : cero o un carácter
- corchetes **[]** : un elemento (de los indicados). Ejemplos:
[abc]: una a, una b o una c
[a-c]: similar al anterior (intervalo)
- corchetes y exclamación **[!]** : Se excluyen los caracteres entre corchetes



Para saber donde se encuentra un comando, ejecutar:

which comando

 Hay elementos que pueden estar en dos localizaciones aunque varíen ligeramente (ej. en /bin y usr/bin)

whereis comando

Muestra la localización de los archivos que tengan relacionados con ese comando

Rutas absolutas vs. rutas relativas

1. Ruta absoluta (desde la raíz /)

/home/luis/Escritorio



Cualquier ruta que comience con una barra inclinada / es una ruta absoluta.

2. Ruta relativa (desde el directorio actual)

Si el directorio actual es /home/luis

Ruta relativa: cd Escritorio/carpeta



Ejemplos de uso de rutas/paths

Rutas absolutas

```
$ cp /home/manz/origen.txt /home/tux/destino.txt
```

Rutas relativas

```
$ cp origen.txt destino.txt
```

Rutas relativas

```
$ cp origen.txt ../tux/destino.txt
```

Ruta absoluta y ruta relativa

```
$ cp /etc/resolv.conf ./backup.conf
```

Standard output (stdout) y standard error (stderr)

Salida estándar (stdout): Es el flujo de salida por defecto donde va la salida normal de un comando. Se muestra en el terminal por defecto. Puede redirigirla a un archivo utilizando >.

Standard error (stderr): Es el flujo de salida para los mensajes de error y las advertencias. También se muestra por defecto en el terminal. Puede redirigirlo a un archivo utilizando >>

Los flujos **stdout** y **stderr** pueden mezclarse a veces en el terminal, pero son canales independientes.



Enlace duros y enlace simbólicos

Enlaces duros (**hardlink**):

Es una referencia directa a un archivo existente (**no funciona con directorios**). Cuando se crea o varios enlaces duros, no se crea una nueva copia de los datos, sino que se incrementa el contador de enlaces del archivo original y **comparten el mismo inodo**. Esto significa que el enlace duro y el archivo original apuntan al mismo contenido en el disco duro. No se pueden crear en diferentes sistemas de archivos. Es creado precediendo por **ln**

```
ln original enlace_duro
```

Características:

- Permiten acceder al mismo archivo o directorio con diferentes nombres.
- No ocupan espacio adicional en el disco duro, ya que solo incrementan el contador de enlaces.
- No se pueden crear en diferentes sistemas de archivos.
- Si se elimina el archivo original, todos los enlaces duros que apuntan a él también se eliminan.

Ejemplos de uso:

- Crear alias (acceso directo) para archivos o directorios de uso frecuente.
- Compartir archivos entre diferentes usuarios sin necesidad de duplicar el contenido.
- Organizar archivos de forma más eficiente.

Enlaces simbólicos (**symlink**):

También conocido como enlace suave o "soft link", es una referencia a un archivo o directorio existente. A diferencia de los enlaces duros, los enlaces simbólicos no apuntan directamente al contenido del archivo, sino que almacenan la ruta del archivo o directorio al que se hace referencia. Puede apuntar a archivos/directorio en otro sistema de archivos diferentes. Se marca con **l** en el tipo de archivo (ver permisos) y una flecha que apunta al original. Es creado precediendo el archivo/directorio origen y destino por **ln -s**

```
ln -s original simbolico
```

⚠ Si al crearlo, *original* no existe, crea el link igual apuntando al *original* ("vacío").

Características:

- Permiten crear referencias a archivos o directorios ubicados en cualquier parte del sistema de archivos, incluso en diferentes sistemas de archivos.
- Ocupan un pequeño espacio en el disco duro para almacenar la ruta del archivo o directorio al que se hace referencia.
- Se pueden crear en cualquier lugar del sistema de archivos.
- Si se elimina el archivo o directorio original, el enlace simbólico sigue existiendo, pero apunta a un archivo o directorio inexistente.
- Cada enlace simbólico tiene su propio número de inodo lo que permite hacer enlaces simbólicos entre distintos sistemas de ficheros.



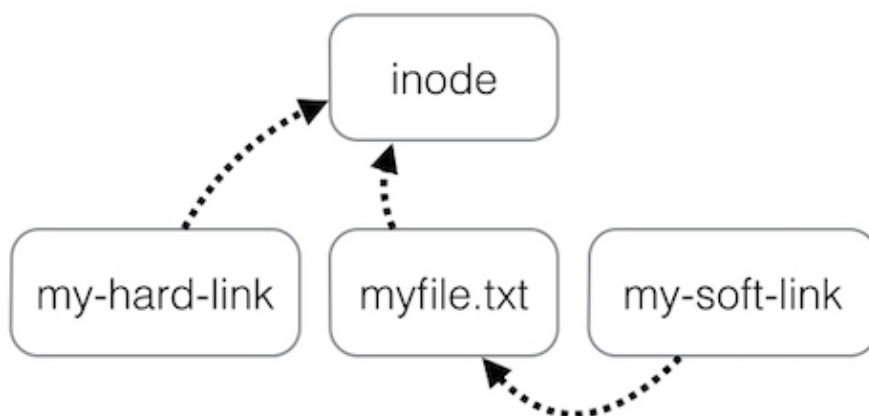
Ejemplos de uso:

- Crear punteros a archivos o directorios que se mueven o cambian de nombre con frecuencia.
- Vincular bibliotecas compartidas en diferentes aplicaciones.
- Montar sistemas de archivos en diferentes ubicaciones.



Los enlaces duros son útiles cuando el archivo original se mueve. Por ejemplo, mover un archivo de `/bin` a `/usr/bin` o a `/usr/local/bin`. Cualquier enlace simbólico al archivo en `/bin` se rompería por esto, pero un enlace duro, siendo un enlace directamente al inodo para el archivo, no le importaría.

Característica	Enlaces duros	Enlaces simbólicos
Definición	Referencia directa a un archivo o directorio	Referencia a la ruta de un archivo o directorio
Ocupación de espacio	No ocupan espacio adicional	Ocupan un pequeño espacio para la ruta
Creación	Solo en el mismo sistema de archivos	En cualquier lugar del sistema de archivos
Eliminación del original	Se eliminan todos los enlaces	El enlace sigue existiendo, pero apunta a un archivo inexistente
Ejemplos de uso	Alias, compartir archivos	Punteros a archivos, bibliotecas, montaje de sistemas de archivos



Para ver los permisos y contador hardlinks

ls -l

Para ver los inodos

ls -li

Para ver hardlink

find . -inum num_inodo

Sudo

SuperUser **do**

 *Switch User: también se puede especificar un usuario no root*

El *superuser* en sistemas antiguos era un usuario real, con un nombre de usuario real (casi siempre "root") con el que podías entrar si tenías la contraseña. Por defecto viene desactivado por defecto

 Pide la contraseña la primera vez, luego la recuerda por ~15min por defecto



sudo sólo puede ejecutar un comando a la vez

No obstante, ese comando podría a su vez ejecutar muchos otros lo que puede generar **problemas de seguridad** si es malicioso

sudo su

Un *truco* con sudo es usarlo para ejecutar el comando **su**. Esto le dará un shell root incluso si la cuenta root está deshabilitada. Puede ser útil cuando necesita ejecutar una serie de comandos como superusuario, para evitar tener que anteponer los sudo a todos, pero abre exactamente el mismo tipo de problemas. Si sigue alguna de las instrucciones que le indican que ejecute sudo su, tenga en cuenta que todos los comandos posteriores se ejecutarán como usuario root. [Fuente](#)

 Más info: <https://help.ubuntu.com/community/RootSudo>

Permisos

El **primer** carácter (de la izquierda) es el tipo de archivo/directorio. Sus valores posibles son los siguientes:

- **-**: Representa un archivo regular.
- **d**: Representa un directorio.
- **l**: Representa un enlace simbólico.
- **c**: Representa un dispositivo de carácter.
- **b**: Representa un dispositivo de bloque.
- **p**: Representa un socket Unix.

Los permisos se agrupan se agrupan de 3 caracteres, correspondiendo a los siguientes *tipos de usuario*:



1. Propietario: El usuario que creó el archivo o directorio.

2. Grupo: Los usuarios que pertenecen al mismo grupo que el propietario.

3. Otros: Todos los demás usuarios del sistema.

Para cada uno de ellos, se pueden otorgar tres permisos básicos:

- **Lectura (r):** Permite ver el contenido de un archivo o listar el contenido de un directorio.
- **Escritura (w):** Permite modificar el contenido de un archivo o agregar/eliminar archivos en un directorio.
- **Ejecución (x):** Permite ejecutar un archivo como un programa o acceder a un directorio.

Cada permiso se representa por un guión (-) o una letra:

- -: Sin permiso.

Esto no aplica en el primer caracter de la izquierda que en ese caso es un archivo regular.

- r: Permiso de lectura.
- w: Permiso de escritura.
- x: Permiso de ejecución.

Los permisos para cada grupo se concatenan y se muestran como una cadena de tres caracteres. Por ejemplo:

- rwx: El propietario tiene permisos de lectura, escritura y ejecución.
- rw-: El grupo tiene permisos de lectura y escritura, pero no de ejecución.
- r--: Otros usuarios solo tienen permiso de lectura.

```
$ ls -l samplefile
```

1	2	3	4	5	6	7
-rw-r--r--	1	pi	pi	5	Nov 29 06:09	samplefile

File Type | Owner Permissions | Group Permissions | Other Permissions | Number of Hard Links | User Owner | Group Owner | File Size in Bytes | Modification Date | File Name

PiMyLifeUp

Ejemplos



- 755: El propietario tiene permisos completos (rwx), el grupo tiene permisos de lectura y ejecución (rw-), y otros usuarios solo tienen permiso de lectura (r--).
- 644: El propietario tiene permisos completos (rwx), el grupo tiene permiso de lectura (r-), y otros usuarios solo tienen permiso de lectura (r--).

Comandos para administrar permisos:

- chmod: Cambia los permisos de archivos y directorios.



[Calculadora online](#)

- chown: Cambia el propietario de un archivo o directorio.
- chgrp: Cambia el grupo propietario de un archivo o directorio.

Understanding The Linux File Permissions

Column: 1 2 3 4

- rwx rwx rwx

Directory/File/Link Information User Rights Group Rights Others Rights

While the first column defines a directory, file or link, the next 3 columns (2, 3, 4) define the permissions for the User, Group and Others (everyone else) groups.

Linux Permissions Made Easy

	user	group	everyone
-	rwx	rwx	rwx
	4 2 1	4 2 1	4 2 1
	1 1 1	1 1 1	1 1 1
	7	7	7

decimal notification
binary notification

decimal notification: add each number to obtain the value (4 + 2 + 1 = 7)
binary notification : convert it to decimal then you should have the value (r-x = 101 base 2 = 5 base 10)



Grupos

Permiten organizar a los usuarios y asignarles permisos de acceso y control sobre archivos, directorios y otros recursos del sistema. Un usuario puede pertenecer a uno o varios grupos, y los permisos de grupo se aplican a todos los miembros del grupo.

Funcionamiento de los grupos:

- **Identificador de grupo (GID):** Cada grupo tiene un identificador único (GID) que lo representa en el sistema.
- **Permisos de grupo:** Los permisos de grupo se establecen para archivos y directorios como se explicó en el punto anterior. Controlan las acciones que los miembros del grupo pueden realizar sobre ellos. Como se observa en la imagen de la anterior página, son los 3 del “medio” (sin contar el primer carácter de la izquierda)
- **Miembros del grupo:** Los usuarios se añaden o eliminan de un grupo utilizando comandos específicos. Los miembros del grupo **heredan** los permisos de grupo para los archivos y directorios a los que tienen acceso.

Creación de grupos:

Para crear un nuevo grupo, se utiliza el comando **groupadd**. La sintaxis básica es la siguiente:

```
sudo groupadd nombre_grupo
```

Donde *nombre_grupo* es el nombre que se desea asignar al nuevo grupo. Por ejemplo, para crear un grupo llamado desarrolladores:

```
sudo groupadd desarrolladores
```

Ver grupos existentes:

Para ver la lista de grupos existentes en el sistema, se utiliza el comando **groups**. Este comando muestra los grupos a los que pertenece el usuario actual.

```
groups
```

Añadir usuarios a un grupo:

Para añadir un usuario a un grupo existente, se utiliza el comando **usermod**. La sintaxis básica es la siguiente:

```
sudo usermod -aG nombre_grupo nombre_usuario
```



Donde `nombre_grupo` es el nombre del grupo al que se desea añadir el usuario y `nombre_usuario` es el nombre del usuario que se desea añadir. Por ejemplo, para añadir el usuario pepe al grupo desarrolladores:

```
sudo usermod -aG desarrolladores pepe
```

Eliminar usuarios de un grupo:

Para eliminar un usuario de un grupo, se utiliza el comando `usermod`. La sintaxis básica es la siguiente:

```
sudo usermod -G nombre_grupo nombre_usuario
```

Donde `nombre_grupo` es el nombre del grupo del que se desea eliminar el usuario y `nombre_usuario` es el nombre del usuario que se desea eliminar. Por ejemplo, para eliminar el usuario pepe del grupo desarrolladores:

```
sudo usermod -G desarrolladores pepe
```

Eliminar un grupo:

Para eliminar un grupo existente, se utiliza el comando `groupdel`. La sintaxis básica es la siguiente:

```
sudo groupdel nombre_grupo
```

Donde `nombre_grupo` es el nombre del grupo que se desea eliminar.

 **Importante:** No se debe eliminar un grupo si hay usuarios que todavía pertenecen a él.



Tubería |

En Linux, una tubería o pipe en inglés (representada por el símbolo de barra vertical **|**) es un mecanismo que permite **conectar la salida de un comando con la entrada de otro comando**. Crea un flujo de datos temporal unidireccional entre los comandos. Permite encadenar eficientemente varios comandos para realizar tareas complejas sin necesidad de guardar resultados intermedios en archivos.

Sintaxis

```
comando1 | comando2 | ... | comandoN
```

comando1: El primer comando que produce una salida.

| El símbolo de tubería que redirige la salida.

comando2: El segundo comando que recibe la salida de comando1 como entrada.

...: Puedes encadenar varios comandos usando tuberías adicionales.

comandoN: El último comando en la tubería que procesa la salida final.

Ejemplos

Listar archivos y filtrar por extensión:

```
ls | grep '.txt' # Lista archivos y muestra solo aquellos con la extensión .txt
```

ls lista todos los archivos en el directorio actual.

La tubería (|) envía la salida (lista de archivos) a grep.

grep '.txt' filtra la lista, mostrando solo las líneas que contienen la cadena .txt.

Contar líneas en un archivo:

```
cat archivo.txt | wc -l # Cuenta el número de líneas en archivo.txt
```

cat archivo.txt lee el contenido del archivo.

La tubería envía el contenido del archivo a wc -l.

wc -l cuenta el número de líneas (-l flag) en los datos recibidos.

Ordenar una lista de archivos:

```
ls | sort # Lista archivos y los ordena alfabéticamente
```

ls lista los archivos.

La tubería envía la lista a sort

sort ordena la lista de archivos alfabéticamente.

Operadores con Tubería

&& Operador "and". Ejecuta varios comandos (si no falla). Ejemplo:



```
test -f archivo.txt && test -r archivo.txt && echo "El archivo archivo.txt existe y es legible"
```

|| Operador "or". Ejecuta el segundo comando solo si falla el 1º

. Indica el directorio actual

.. Indica el directorio superior en el árbol

!./ Ejecuta el último programa

Consejos adicionales

- Puedes encadenar varias tuberías para crear flujos de trabajo de procesamiento más complejos.
- Usa la redirección (>, >>) para guardar la salida final de la tubería en un archivo.
- Explora las opciones de varios comandos (flags) para refinar aún más el comportamiento de cada comando en la tubería.

TTY y terminales virtuales

TTY en Linux significa "TeleTYpewriter" (Teletipo) y se refiere a un dispositivo que permite la comunicación texto a texto. En el contexto de Linux, un TTY normalmente representa una terminal virtual, que es una sesión de línea de comandos (CLI).

Terminales virtuales: Linux puede tener múltiples terminales virtuales activas simultáneamente. Cada terminal virtual se asigna a un TTY específico, como tty1, tty2, etc. Puedes cambiar entre estas terminales virtuales usando la combinación de teclas Ctrl + Alt + F seguido del número de la terminal (por ejemplo, Ctrl + Alt + F2).

Número de terminal cambian según usuario. Compruébalo haciendo login con más de un user

Instalación de paquetes/aplicaciones

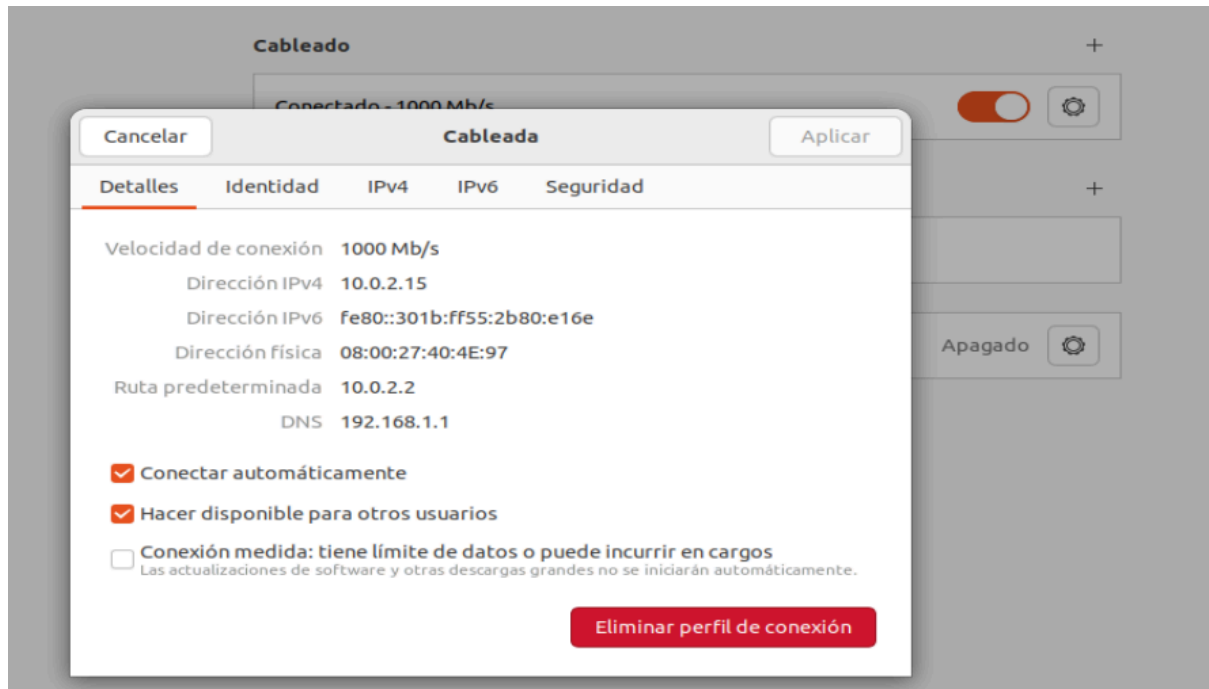
apt
apt-get

TODO: Añadir ejemplos

Networking

Configuración

- GUI → Red





Terminal

- net-tools → **Deprecated**

Aún se pueden utilizar pero requieren instalación y **no es recomendable** por no estar mantenidos.

Comando shell

Deprecated	Reemplazo
arp	ip n (ip neighbor)
ifconfig	ip a (ip addr), ip link, ip -s (ip -stats)
iptunnel	ip tunnel
iwconfig	iw
nameif	ip link, ifrename (requiere instalación)
netstat	ss ip route (equivale a netstat -r) ip -s link (equivale a netstat -i) ip maddr (equivale a netstat -g)
route	ip r (ip route)

- Compartir en red
 - scp (ssh)
 - rsync
 - HTTP Server
 - NFS
 - Samba
- Host table /etc/hosts
- Cómo renovar (flush) la IP

<https://www.cyberciti.biz/faq/linux-ip-command-examples-usage-syntax/>
<https://itsubuntu.com/50-useful-linux-networking-commands-and-tools/>



Scripting

Diferencia entre comillas simple y dobles

- Comillas simples: Todo lo que va entre comillas simples se trata como una cadena literal. Los nombres de variables y la mayoría de los caracteres especiales no se expanden.
- Comillas dobles: Las variables y ciertos caracteres especiales entre comillas dobles se expanden. El contenido está sujeto a sustitución de variables y sustitución de comandos.

Para que un script se **ejecute** debe tener los **permisos** necesarios

```
chmod +x nombreScript.sh
```

Esto se debe realizar para todos los scripts

```
#!/bin/bash # Que interprete tiene que correr el script
```

```
nombreVariable="Hola"
```

```
echo '$nombreVariable' # Salida: $nombreVariable
```

```
echo "$nombreVariable" # Salida: Hola
```

Para ejecutarlo

```
./nombreScript
```

asumiendo que está en su misma ruta, sino podríamos usar rutas relativas o absolutas



Un script bash puede terminarse pulsando CTRL + C
sin esperar a que termine su operación.

<https://www.freecodecamp.org/news/shell-scripting-crash-course-how-to-write-bash-scripts-in-linux/>

<https://www.geeksforgeeks.org/shell-script-examples/>

<https://www.howtogeek.com/808593/bash-script-examples/>

<https://www.hostinger.com/tutorials/bash-script-example>



WSL

WSL (Windows Subsystem for Linux) es una capa de virtualización de Linux que permite ejecutar algunas distribuciones de Linux sin la necesidad de instalar una máquina virtual o dual boot. En el proceso de instalación se debe elegir la distribución y versión como se verá después (aunque viene una versión por defecto). Por ello requiere de muchos menos recursos.

Instalación

En **Powershell** o cmd

```
wsl --install
```

<https://terminaldelinux.com/terminal/wsl/instalacion-wsl/>

<https://es.linkedin.com/pulse/c%C3%B3mo-instalar-wsl2-en-windows-10-subsystem-linux-versi%C3%B3n-2-employit>

<https://learn.microsoft.com/es-es/windows/wsl/install>

🔥 Cuando termine la instalación tras reiniciar te pedirá que crees el usuario y password.

💻 Para ejecutarlo (ambas funcionan)

- Buscar "ubuntu" (o la distribución instalada) en el buscador de windows
 - Ejecutar **wsl** en cmd/powershell
-

Ruta

- Para interacción con archivos Linux files directamente desde Windows:

\\wsl\$\nombreDistribucion

- Para acceso network-related para la distribución de WSL desde Windows.

\\wsl.localhost\nombreDistribucion

Instalar GUI software para WSL2

<https://learn.microsoft.com/en-us/windows/wsl/tutorials/gui-apps>

https://techviewleo.com/run-linux-gui-applications-on-windows/?utm_content=cmp-true

WSL1 vs. WSL2


La diferencia principal entre WSL 1 y WSL 2 es que este último opera dentro de una máquina virtual administrada (VM).

Por lo general, Microsoft recomienda utilizar WSL 2 debido a su rendimiento superior en comparación con WSL 1 y a su compatibilidad al 100% con las llamadas al sistema.



No obstante, se puede utilizar WSL 1 si un proyecto debe almacenarse en el sistema de archivos de Windows o si se requiere la compilación cruzada (cross-compilation) con herramientas de Windows y Linux para un proyecto.

 <https://learn.microsoft.com/en-us/windows/wsl/compare-versions>

 <https://tuxcare.com/blog/what-is-windows-subsystem-for-linux/#:~:text=WSL%20has%20some%20limitations%2C%20however,file%20systems%2C%20such%20as%20ext4.>

WSL Vs Cygwin

https://www.reddit.com/r/linux/comments/10x8mcw/is_there_any_advantage_to_wsl_over_cygwin

Cygwin y Miygwin

- Compilas aplicaciones nativas de Windows.



Otros

Instalar guest additions en Virtual Box

sudo apt-get install virtualbox-guest-utils
O ejecutar el .sh desde el CD ROM Virtual

Libro Linux

<https://archive.org/details/la-linea-de-comandos-de-linux/page/n1/mode/1up>



Tutoriales y juegos aprendizaje Bash

Tutoriales

- <https://linuxjourney.com/>
- <https://linuxsurvival.com/>

Juegos

- [command line heroes bash](#)
- <https://overthewire.org/wargames/>
- <https://trybash.github.io/game/>

problema resolucion?



Glosario de términos

apt

apt-get

Boot loader: GRUB o ISOLINUX

Daemon (Demonio)

GRUB: cargador de arranque

File system (sistema de archivos): método de guardado y organización de archivos

Inodo

Make → Makefile

[Número Mágico](#) (no confundir el mismo término usado en programación)

Packet Manager

Servicio: Programa que se ejecuta como proceso en el background

Shell: bash, tcsh, zsh