

# Intro a la Terminal de Comandos de Linux

Diana García Cortés

# Sistema Operativo

# Sistema Operativo

Software principal, que gestiona los recursos de hardware, se ejecuta en modo privilegiado y provee servicios a las aplicaciones.

# Software Libre

Respetar la libertad de los usuarios y la comunidad.  
Los usuarios tienen la libertad de ejecutar, copiar, distribuir,  
estudiar, modificar y mejorar el software.

Free Software Foundation

# Software Libre

Respetar la libertad de los usuarios y la comunidad.

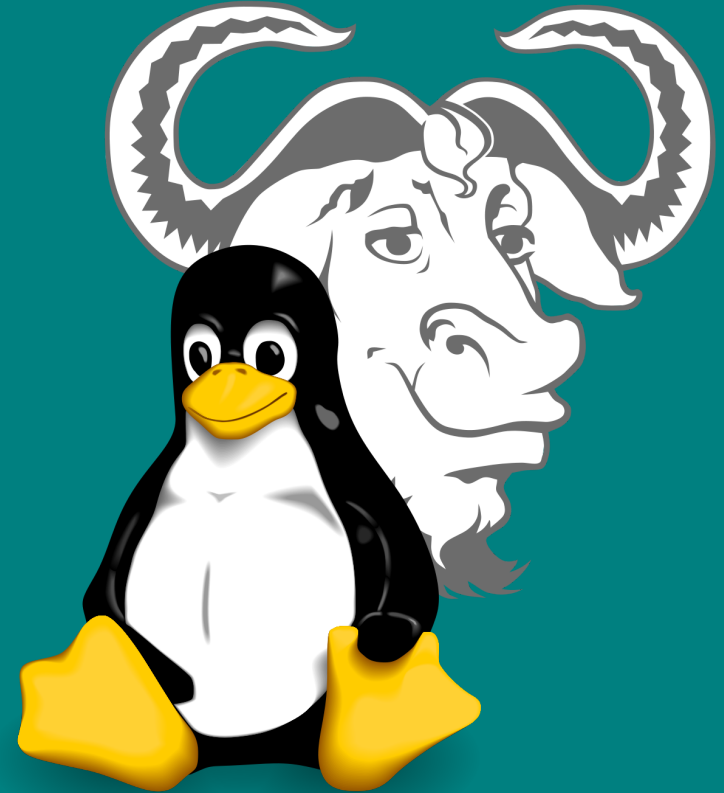
Los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

0. Libertad de ejecutar el programa como se desee, con cualquier propósito
1. Libertad de acceso al código fuente: posibilidad de estudiar y modificar el programa.
2. Libertad de redistribuir copias para ayudar a otros.
3. Libertad de distribuir copias de sus versiones modificadas a terceros.

Free Software Foundation

# Proyecto GNU/Linux

- Sistema Operativo completamente de software libre.
- Las herramientas son GNU, el kernel es Linux.
- Proyecto que inició desde 1984.
- Un sistema basado en Unix.



## Algunos conceptos importantes

- Distribuciones
- Sistemas de archivos
- Manejadores de paquetes

# Distribución GNU/Linux

Sistema Operativo con el kernel de Linux, herramientas y librerías de GNU, software adicional, sistema y manejador de ventanas y un entorno de escritorio.

**Diferencias:** Instalador utilizado, manejador de paquetes, aplicaciones incluidas, entorno de escritorio.

¿Qué distros conocen?



# Sistema de archivos

Conjunto de normas y procedimientos para almacenar la información.

Sistema de almacenamiento de un dispositivo de memoria, que estructura y organiza la escritura, búsqueda, lectura, almacenamiento, edición y eliminación de archivos de una manera concreta.

- **FAT (File Allocation Table o tabla de asignación de archivos)**  
(4GB - 8TB). Volumen de datos pequeño. USB y dispositivos móviles.
- **NTFS (New Technology File System)**  
Estándar para Windows desde Windows Vista.
- **APFS (Apple File System)**  
Apple en 2017 con High Sierra para unidades de estado sólido modernas. Siguiendo la generación de HFS+.
- **ext4**  
Estándar para sistemas Linux. Optimiza la gestión de archivos grandes y evita la fragmentación de manera más eficaz. Las particiones se pueden ampliar y reducir según sea necesario, e incluso durante el procesamiento. Volumen de 1 exabyte (aproximadamente 1 millón de terabytes).

## ¿Por qué nos importan los sistemas de archivos?

- Todos los sistemas operativos pueden leer y escribir en USBs (FAT)
- ¿Han instalado Linux en una partición? ¿Windows puede ver los archivos de Linux?

## ¿Por qué nos importan los sistemas de archivos?

- Todos los sistemas operativos pueden leer y escribir en USBs (FAT)
- ¿Han instalado Linux en una partición? ¿Windows puede ver los archivos de Linux?
  - NTFS no puede leer y/o escribir ext4

# Filesystem Hierarchy Standard or FHS

Filesystem Hierarchy Standard (FHS) is a set of guidelines that defines the structure and organization of the Linux filesystem hierarchy.

The FHS defines the standard locations for system files, user files, and application files, ensuring consistency across different Linux distributions.

The FHS is maintained by the Linux Foundation and is widely adopted by most Linux distributions.

The FHS defines the following hierarchy:

- / (root directory)
- /bin (binary executables)
- /boot (boot files)
- /dev (device files)
- /etc (configuration files)
- /home (user home directories)
- /lib (libraries)
- /media (mountable file systems)
- /mnt (mount point for temporary file systems)
- /opt (optional software packages)
- /sbin (system binaries)
- /srv (server data)
- /tmp (temporary files)
- /usr (user files)
- /var (variable files)

The FHS also defines the standard locations for system files, user files, and application files.

The FHS is a critical component of the Linux operating system, ensuring that all Linux distributions follow the same standards.

The FHS is a set of guidelines that defines the structure and organization of the Linux filesystem hierarchy.

Nació en 1994, mantenido por la Linux Foundation.



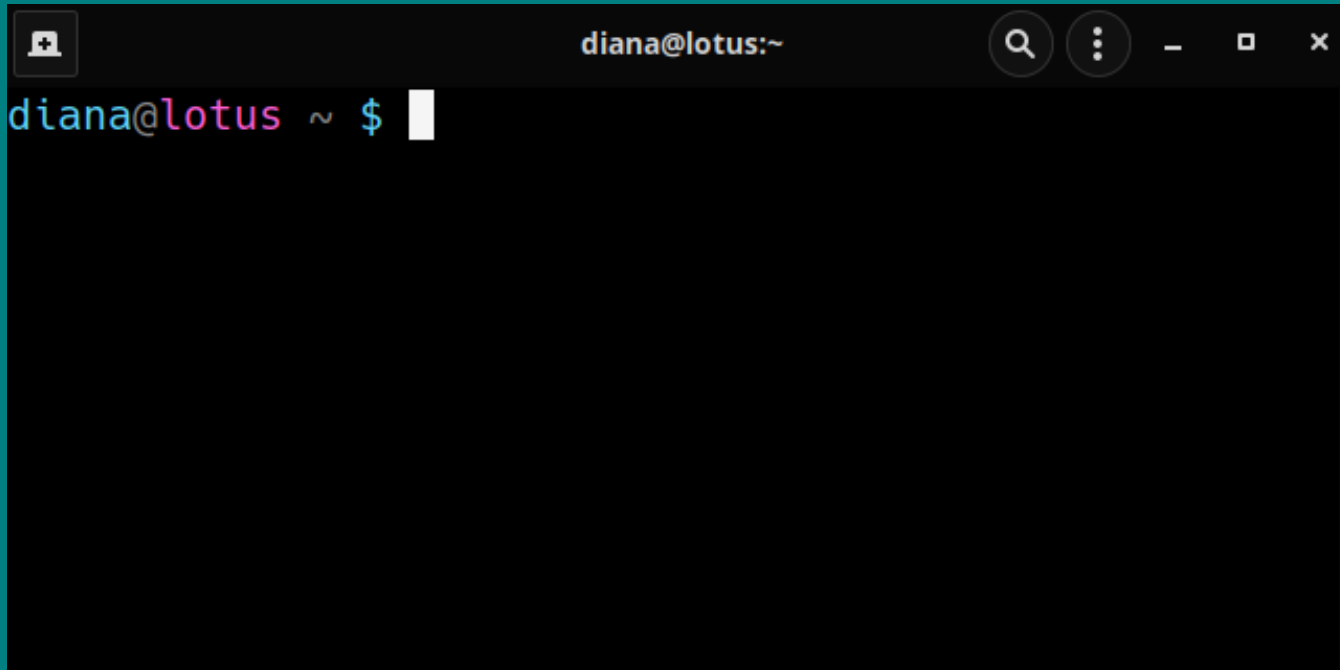
|       |   |
|-------|---|
| bin   | Binarios de usuario   |
| boot  | Ejecutables y archivos requeridos para el arranque                    |
| dev   | Archivos de información de todos los volúmenes                        |
| etc   | Archivos de configuración del sistema y de aplicaciones               |
| home  | Directorio personal con las carpetas de usuario                       |
| lib   | Bibliotecas necesarias para la ejecución de binarios                  |
| media | Directorio de montaje de volúmenes extraíbles                         |
| opt   | Ficheros de aplicaciones externas que no se integran en /usr          |
| proc  | Ficheros de información de procesos                                   |
| root  | Directorio personal de superusuario                                   |
| sbin  | Binarios de sistema   |
| srv   | Archivos relativos a servidores web, FTP, etc.                        |
| sys   | Archivos virtuales con información de eventos del sistema             |
| tmp   | Directorio de ficheros temporales                                     |
| usr   | Archivos de programas y aplicaciones instaladas                       |
| var   | Archivos de variables, logs, emails de los usuarios del sistema, etc. |

# Manejador de paquetes

Colección de herramientas de software que automatizan el proceso de instalación, actualización, configuración y eliminación de programas para un SO.

| Distribución   | Manejador de paquetes | Herramientas           |
|--|-----------------------|------------------------|
| Debian, Ubuntu   | dpkg                  | apt, aptitude          |
| Red Hat, open Suse, Fedora                             | RPM                   | apt4rm, zypp, yum, dnf |
| Arch Linux, Frugalware                                 | Pacman                | Makepkg                |
| Knoppix, Mint, Gentoo, OpenSUSE, Red Hat and Slackware | Zero Install          | 0install               |

# Conozcamos la terminal



Este es el prompt, listo para ejecutar instrucciones.



# Conozcamos la terminal

Este es el prompt

- diana:
- lotus:
- ~:
- \$:

# Conozcamos la terminal

Este es el prompt

- `diana: usuario`
- `lotus:`
- `~:`
- `$:`

# Conozcamos la terminal

Este es el prompt

- `diana: usuario`
- `lotus: hostname`
- `~:`
- `$:`

# Conozcamos la terminal

Este es el prompt

- **diana:** usuario
- **lotus:** hostname
- **~:** la ruta (directorio) donde estamos
- **\$:**

# Conozcamos la terminal

Este es el prompt

- **diana:** usuario
- **lotus:** hostname
- **~:** la ruta (directorio) donde estamos
- **\$:** significa que nuestro intérprete de comandos o shell, es bash o GNU Bourne-Again shell

## Estructura de los comandos

```
commando [opciones] [arg1] [arg2] ...  
          [argn]
```

Las opciones y los argumentos dependen del  
comando que estemos ejecutando

# Calentando

## Comandos que no necesitan argumentos

- 

```
1 pwd
```

- 

```
1 whoami
```

- 

```
1 hostname
```

- 

```
1 date
```

# Calentando

Comandos que no necesitan argumentos

- Print working directory. Dónde estamos

```
1 pwd
```

- 

```
1 whoami
```

- 

```
1 hostname
```

- 

```
1 date
```



# Calentando

Comandos que no necesitan argumentos

- Print working directory. Dónde estamos

```
1 pwd
```

- Nombre de usuario loggeado en la consola

```
1 whoami
```

- 

```
1 hostname
```

- 

```
1 date
```

# Calentando

Comandos que no necesitan argumentos

- Print working directory. Dónde estamos

```
1 pwd
```

- Nombre de usuario loggeado en la consola

```
1 whoami
```

- El nombre del equipo

```
1 hostname
```

- 

```
1 date
```

# Calentando

Comandos que no necesitan argumentos

- Print working directory. Dónde estamos

```
1 pwd
```

- Nombre de usuario loggeado en la consola

```
1 whoami
```

- El nombre del equipo

```
1 hostname
```

- Fecha y hora en formato largo

```
1 date
```

# Calentando

## Comandos que no necesitan argumentos

- 

```
1 env
```

- 

```
1 who
```

- 

```
1 uname -a
```

# Calentando

Comandos que no necesitan argumentos

- Muestra las variables de entorno

```
1 env
```

- 

```
1 who
```

- 

```
1 uname -a
```

# Calentando

Comandos que no necesitan argumentos

- Muestra las variables de entorno

```
1 env
```

- Info sobre nuestra sesión

```
1 who
```

- 

```
1 uname -a
```

# Calentando

Comandos que no necesitan argumentos

- Muestra las variables de entorno

```
1 env
```

- Info sobre nuestra sesión

```
1 who
```

- Info sobre nuestro sistema

```
1 uname -a
```

Por default, nuestra shell imprime el resultado en la consola, se le llama **stdout** o standard output stream. Para cambiarlo:

```
1 uname -a > sesion.txt
```

Esto se llama redirigir la salida.



## Archivos texto plano

Solamente contienen texto. No Word.

Extensiones: .txt, .tsv, .csv, etc

## Archivos texto plano

Solamente contienen texto. No Word.

Extensiones: .txt, .tsv, .csv, etc

Para verlos:

## Archivos texto plano

Solamente contienen texto. No Word.

Extensiones: .txt, .tsv, .csv, etc

Para verlos:

```
1 cat data_rentas.tsv  
2 more data_rentas.tsv  
3 less -S data_rentas.tsv
```

# Comandos para manejar directorios y archivos

## Listar los contenidos de un directorio

```
1 ls
2 ls -a
3 ls -al
4 ls -alh
5 ls -alht
6 ls -alht /bin
```

## ¿Cómo vemos todas las opciones?

```
1 man ls
2 ls --help
```

# Hay carpetas que siempre existen:

• ■

• ■ ■

• ~

# Hay carpetas que siempre existen:





- 

El directorio actual

-  

- 

# Hay carpetas que siempre existen:

-  El directorio actual
-   Un directorio atrás
- 

# Hay carpetas que siempre existen:

- - El directorio actual
- - .. Un directorio atrás
- ~ Nuestra carpeta **\$HOME**



Hay carpetas que siempre existen:

- `.`  
El directorio actual
- `..`  
Un directorio atrás
- `~`  
Nuestra carpeta `$HOME`

En realidad no existen, son referencias a esas carpetas.

## Un poco sobre rutas

Hay dos tipos

- Rutas relativa: define un archivo o directorio a partir del directorio donde nos encontramos.
- Rutas absoluta: define de forma única un archivo o directorio. Siempre inician con /, desde la raíz.

# Comandos para manejar directorios y archivos

## Cambiarnos de directorio

```
1 cd /home/[tu_username]/Desktop  
2 cd ../Documents  
3 cd ../../../../tmp
```

Utilicen siempre el **TAB** y practiquen mucho

# Comandos para manejar directorios y archivos

## Crear directorios o carpetas

```
1 cd ~/Desktop
2 mkdir carpeta1
3 mkdir carpeta1 carpeta2
```

## También se vale adentro de otras

```
1 cd ~/Desktop
2 mkdir ../Documents/padre
3 mkdir ../Documents/padre/hijo
4 mkdir -p ../Documents/madre/hija
```

# Comandos para manejar directorios y archivos

## Copiar y mover archivos y carpetas

```
1 cd ~/Desktop
2 touch nada.txt
3 cp nada.txt ../Documents/
4 mv nada.txt ../Documents/algo.txt
```

## Para carpetas tiene que ser recursivo

```
1 cd ~/Desktop
2 mkdir carpeta3
3 touch carpeta3/archivo.txt
4 cp -r carpeta3 ../Documents/carpeta_copia
5 mv -r ../Documents/carpeta_copia carpeta_nueva
```

# Comandos para manejar directorios y archivos

## Borrar archivos y carpetas

OJO! No hay papelera de reciclaje

```
1 rm carpeta_nueva/archivo.txt  
2 rmdir carpeta_nueva
```

# Comandos para manejar directorios y archivos

Borrar archivos y carpetas

OJO! No hay papelera de reciclaje

```
1 rm carpeta_nueva/archivo.txt  
2 rmdir carpeta_nueva
```

Modo salvaje

# Comandos para manejar directorios y archivos

## Borrar archivos y carpetas

OJO! No hay papelera de reciclaje

```
1 rm carpeta_nueva/archivo.txt  
2 rmdir carpeta_nueva
```

## Modo salvaje

```
1 rm -rf carpeta3
```



## Ejercicio

1. Ve a tu `$HOME`
2. Crea un directorio con tu mes favorito del año.
3. Sin moverte de directorio, guarda la salida del comando `date` en un archivo llamado `hoy.txt` dentro de la carpeta que acabas de crear.
4. Sin moverte de directorio, copia la carpeta a tu escritorio con el nombre de tu animal favorito.
5. Verifica que la carpeta haya sido creada.
6. Sin moverte de directorio, mueve el archivo en `~/Desktop/[animal favorito]/hoy.txt` a la carpeta de tu mes favorito con el nombre `ejercicio.txt`
7. Borra las carpetas y los archivos que creamos.

Descarguen el siguiente archivo comprimido en zip:  
**WWC.zip**

O pueden descargar el .tar.xz desde la terminal

```
1 wget https://www.dropbox.com/s/j7syscpyu35ucpn/WWC.tar.xz?dl=0
2 tar -xvf WWC.tar.xz
```

# grep

Nos permite buscar texto en archivos de **texto plano**.

Buscar una palabra en un archivo

```
1 grep arroz arroz_leche.txt
```

Buscar una palabra sin importar mayúsculas

```
1 grep -i Manzana tarta_manzana.txt
```

## Con el número de línea y colores

```
1 grep -in --color calabaza mousse_calabaza.txt
```

## Buscar la palabra completa

```
1 grep -inw Manzana --color tarta_manzana.txt
```

Si solo queremos contar cuántas veces aparece la palabra.

```
1 grep -c calabaza bund_cake_calabaza.txt
```

Si queremos contar en todos los archivos del directorio

```
1 grep -Rc canela .
```

O ver las líneas completas

```
1 grep -Rni --color canela .
```

## Ejercicio

- ¿Cuántas películas del archivo `data_rentas.tsv` fueron rentadas por mujeres?
- ¿Por cuántos días rentaron la película Coyote Ugly?
- ¿Cuántas veces rentaron Iron Man?
- ¿Cuántas películas tienen "The" en su título?

# Un poco más sobre archivos de texto tabulares

Para saber cuántas líneas y cuántas palabras tiene un archivo

```
1 wc data_rentas.tsv  
2 wc -l data_rentas.tsv
```

Linux es muy bueno para trabajar con archivos tabulares

```
1 cut -f 5 data_rentas.tsv  
2 cut -f 5 data_rentas.tsv | less
```

Regresando al ejercicio: ¿Cuántas películas tienen  
"The" en su título?



Regresando al ejercicio: ¿Cuántas películas tienen "The" en su título?

```
1 grep -w The data_rentas.tsv
```

Regresando al ejercicio: ¿Cuántas películas tienen "The" en su título?

```
1 grep -w The data_rentas.tsv
```

```
1 cut -f 5 data_rentas.tsv | grep -w The
```

# find

Sirve para encontrar archivos en nuestra computadora

Por nombre

```
1 find . -name iris.txt
```

El `.` representa el directorio en el cual queremos buscar el archivo

Todas las imágenes .jpg en nuestro \$HOME

```
1 find /home/diana -name *.jpg
```

Las que fueron modificadas en los últimos 3 días

```
1 find /home/diana -name "*jpg" -mtime 3
```

También podemos buscar directorios

```
1 find /home/diana -name WWC -type d
```

**Extras**

# Aliases

Sobrenombres para comandos que son largos, a veces se nos olvidan o los utilizamos mucho.

Creamos un archivo llamado `.bash_aliases` en nuestro directorio `$HOME`

```
1 alias ll="ls -lhta"  
2 alias cl-out="ssh -p 5263 -X dgarcia@cluster.inmegen.gob.mx"
```

Cerramos y abrimos la consola o escribimos:

```
1 source ~/.bash_aliases
```

# Procesos

El linux cada tarea que se esté ejecutando es un proceso, que puede distinguirse por su process ID o PID.

```
1 ps aux
2 ps aux | grep chrome
3 htop
```

Si un proceso deja de responder, tiene que morir

```
1 kill PID
```



Si un proceso deja de responder, tiene que morir

```
1 kill PID
```

Y si se rehúsa, lo forzamos

Si un proceso deja de responder, tiene que morir

```
1 kill PID
```

Y si se rehúsa, lo forzamos

```
1 kill -9 PID
```

# Usuarios en Linux

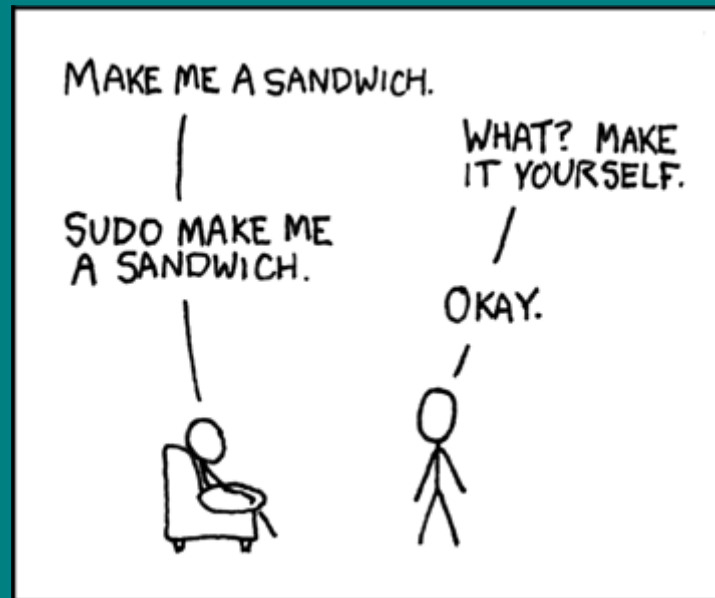
Todos tienen UserID (UID) y pertenecen, al menos, a un grupo de usuario, con GroupID (GID) único.

# Usuarios en Linux

Todos tienen UserID (UID) y pertenecen, al menos, a un grupo de usuario, con GroupID (GID) único.

- **Usuario root, superusuario, administrador:** UID=0. Privilegios sobre todo el sistema, acceso a todos los archivos y directorios.
- **Usuarios especiales:** Asociados a cuentas o servicios del sistema, no tienen inicio de sesión, se general al instalar Linux o su aplicación asociada. UID entre 1 y 100.
- **Usuarios normales:** Normalmente tienen directorio de trabajo, privilegios solo en su directorio de trabajo o HOME, UID superior a 500.

# xkcd



# Permisos

## Tres tipos de permisos

- **Lectura (read):** Ver contenido de un archivo o directorio.
- **Escritura (write):** Modificar el archivo. Agregar, sobrescribir o borrar en archivo o directorio
- **Execución (execute):** Ejecutar un archivo. En directorios, acceder a él y permitir que otros programas puedan acceder a él

Tres tipos de acceso: user (u), group (g), others(o)

Cambiamos los permisos con el comando **chmod**

Damos permisos de ejecución al usuario

```
1 chmod u+x hello.sh
```

Quitamos permisos de lectura al grupo

```
1 chmod g-r hello.sh
```

**Mas sobre archivos tabulados**



**Gracias!**