

Robin Jack Schilmöller

## Práctica 2: Administración básica de Linux

(Sistemas Informáticos – 1º DAM, Curso 2025/2026 – Albor Croft – ILERNA)

### Parte 1: Repaso de comandos básicos (Práctica 1)

1. Crea una carpeta llamada repaso dentro de tu carpeta personal.

```
$ cd ~/
```

```
$ mkdir repaso
```

2. Dentro de repaso:

- Crea un archivo vacío.

```
$ touch archivo_vacio
```

- Hazle una copia con otro nombre (cp).

```
$ cp archivo_vacio archivo_copia
```

- Renombra la copia (mv).

```
$ mv archivo_copia archivo_renombrado
```

- Borra uno de los archivos (rm).

```
$ rm archivo_renombrado
```

3. Crea dentro de repaso una subcarpeta llamada pruebas y mueve un archivo dentro de ella (mkdir, mv).

```
$ mkdir pruebas
```

```
$ mv archivo_vacio pruebas/
```

4. Copia la carpeta pruebas completa a otra ubicación (cp -r).

```
$ mv -r pruebas ~/
```

5. Mueve un archivo con confirmación antes de sobrescribir (mv -i).

```
jack@Ubuntu:~$ touch archivo archivo2
jack@Ubuntu:~$ mv archivo pruebas/
jack@Ubuntu:~$ mv -i archivo2 pruebas/archivo
mv: overwrite 'pruebas/archivo'? yes
jack@Ubuntu:~$
```

6. Borra un directorio vacío (rmdir).

```
$ mkdir dir_vacio
```

```
$ rmdir dir_vacio
```

7. Borra una carpeta con todos sus archivos dentro (rm -r).

```
$ rm -r pruebas
```

8. Muestra el contenido de /etc/passwd en pantalla (cat).

```
$ cat /etc/passwd
```

9. Muestra solo las primeras 5 líneas de ese archivo (head -n).

```
$ head -5 /etc/passwd
```

**10. Lista todos los archivos, incluidos los ocultos, de tu carpeta personal (ls -a).**

```
$ ls -a ~/
```

**11. Muestra el contenido detallado (permisos, tamaño, fecha) de un directorio (ls -l).**

```
$ ls -la ~/
```

**12. Consulta:**

- Tu usuario actual (whoami).

```
$ whoami
```

- Tu directorio de trabajo (pwd).

```
$ pwd
```

- El espacio libre en disco (df -h).

```
$ df -h
```

## Parte 2: Usuarios y grupos

**1. Crea un nuevo usuario llamado admin1.**

```
$ sudo useradd admin1
```

>> y escribir contraseña para admin

```
$ sudo usermod -aG sudo admin1
```

**2. Crea otro usuario llamado alumno1.**

```
$ sudo useradd alumno1
```

>> y escribir contraseña para alumno

**3. Desde admin1, crea un grupo llamado alumnos.**

```
$ su admin1
```

>> escribir contraseña admin

```
$ sudo groupadd alumnos
```

**4. Añade al usuario alumno1 dentro del grupo alumnos.**

```
$ sudo usermod -aG alumnos alumno1
```

**5. Cambia la contraseña de alumno1.**

```
$ sudo passwd alumno1
```

>> escribir contraseña admin

**6. Cambia el nombre de tu usuario inicial (el primero que usaste en Ubuntu).**

```
$ sudo usermod -l jack2 jack
```

Pregunta de reflexión: ¿Qué riesgos tendría que cualquiera pudiera cambiar contraseñas de otros usuarios?

>> Es un riesgo de seguridad, un usuario podría tomar control (y a la vez quitárselo a esos usuarios) de otras cuentas al cambiar la contraseña.

### Parte 3: Instalación de programas

1. Desde el usuario admin1, actualiza la lista de paquetes del sistema.

```
$ sudo apt-get update
```

2. Instala Python 3 y comprueba su versión.

```
$ sudo apt-get install python3
```

```
$ python3 -V
```

3. Instala Sublime Text siguiendo el enlace oficial:

👉 [https://www.sublimetext.com/docs/linux\\_repositories.html](https://www.sublimetext.com/docs/linux_repositories.html)

```
$ wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install sublime-text
```

4. Instala el comando tree y pruébalo dentro de tu carpeta personal para visualizar su estructura de directorios.

```
$ sudo apt-get install tree
```

```
$ tree ~/
```

Pregunta de reflexión: ¿Por qué se necesita sudo para instalar programas?

>> Por seguridad, si todos los usuarios un usuario con malas intenciones (o no) podría instalar algún programa que dañe el sistema.

### Parte 4: Permisos en Linux

1. Desde admin1, crea una carpeta llamada claseSI dentro de tu carpeta personal.

```
$ cd ~/
```

```
$ mkdir claseSI
```

2. Dentro de claseSI, crea un archivo instrucciones.txt con un texto de prueba.

```
$ nano claseSI/instrucciones.txt
```

>> escribir texto dentro del archivo

3. Configura los permisos de forma que:

- El usuario admin tenga lectura y escritura.
- El grupo alumnos tenga solo lectura.
- Otros usuarios no tengan acceso.

```
$ chmod 640
```

```
$ usermod -aG admin1 alumno1
```

>> añadimos al usuario alumno1 al grupo de admin1 para que pueda leer el archivo

4. Inicia sesión como alumno1 e intenta leer el archivo.

```
$ su alumno1
```

```
$ cat /home/admin1/instrucciones.txt
```

5. Intenta modificarlo y explica qué sucede.

>> No es posible ya que únicamente tiene permisos de lectura

```
[ File 'instrucciones.txt' is unwritable ]
^M Where Is ^M Cut ^M Execute
```

Pregunta de reflexión: ¿Por qué es importante que ciertos archivos no sean modificables por todos los usuarios?

>> Porque hay archivos como /etc/passwd que contiene contraseñas del sistema, si todos los usuarios pudieran cambiar las contraseñas, el sistema no tendría seguridad ninguna.

## Parte 5: Búsquedas con find, grep y locate

- ♦ Ejercicios con find

### 1. Busca todos los archivos que terminen en .txt en tu carpeta personal.

👉 Pista: find [ruta] -name "\*.txt"

```
jack@Ubuntu:~$ find . -name "*.txt"
./DAM1/redes/robinjack_dam_copia.txt
./DAM1/robinjack_dam_26_09_2025.txt
./.cache/tracker3/files/last-crawl.txt
./.cache/tracker3/files/first-index.txt
```

### 2. Busca todos los directorios dentro de tu carpeta personal.

👉 Pista: find [ruta] -type d

```
jack@Ubuntu:~$ find . -type d
.
./Documents
./DAM1
./DAM1/LMGSI
./DAM1/redes
```

### 3. Busca archivos que tengan exactamente el nombre notas.txt.

👉 Pista: find [ruta] -name "notas.txt"

```
jack@Ubuntu:~$ find . -name "notas.txt"
jack@Ubuntu:~$
```

>> no existe el archivo notas.txt

### 4. Busca archivos que se hayan modificado en las últimas 24 horas.

👉 Pista: find [ruta] -mtime -1

```
jack@Ubuntu:~$ find . -mtime -1
./.local/state/wireplumber
./.local/state/wireplumber/restore-stream
./.local/share/keyrings
```

### 5. Busca todos los archivos de más de 1 MB en tu carpeta personal.

👉 Pista: find [ruta] -size +1M

```
jack@Ubuntu:~$ find . -size +1M
./.cache/mesa_shader_cache_db/index
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%23Pictures.db
./.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2
```

- ♦ Ejercicios con grep

## 6. Busca qué archivos contienen la palabra Linux en tu carpeta personal.

👉 Pista: grep -r "Linux" [ruta]

```
jack@Ubuntu:~$ grep -r "linux" .
./snap/firmware-updater/common/.cache/gdk-pixbuf-loaders.cache:# LoaderDir =
./snap/firmware-updater/167/gnome-platform/usr/lib/x86_64-linux-gnu/gdk-pixb
uf-2.0/2.10.0/loaders
./snap/firmware-updater/common/.cache/gdk-pixbuf-loaders.cache:"/snap/firmwa
re-updater/167/gnome-platform/usr/lib/x86_64-linux-gnu/gdk-pixbuf-2.0/2.10.0
/loaders/libpixbufloader-ani.so"
```

## 7. Haz la misma búsqueda pero ignorando mayúsculas/minúsculas.

👉 Pista: grep -i "Linux" archivo.txt

```
jack@Ubuntu:~$ grep -ri "linux" .
./snap/firmware-updater/common/.cache/gdk-pixbuf-loaders.cache:# LoaderDir =
./snap/firmware-updater/167/gnome-platform/usr/lib/x86_64-linux-gnu/gdk-pixb
uf-2.0/2.10.0/loaders
./snap/firmware-updater/common/.cache/gdk-pixbuf-loaders.cache:"/snap/firmwa
re-updater/167/gnome-platform/usr/lib/x86_64-linux-gnu/gdk-pixbuf-2.0/2.10.0
/loaders/libpixbufloader-ani.so"
```

## 8. Muestra también el número de línea en el que aparece.

👉 Pista: grep -n "Linux" archivo.txt

```
jack@Ubuntu:~$ grep -rin "linux" .
./snap/firmware-updater/common/.cache/gdk-pixbuf-loaders.cache:5:# LoaderDir
= /snap/firmware-updater/167/gnome-platform/usr/lib/x86_64-linux-gnu/gdk-pi
xbuf-2.0/2.10.0/loaders
./snap/firmware-updater/common/.cache/gdk-pixbuf-loaders.cache:7:"/snap/firm
ware-updater/167/gnome-platform/usr/lib/x86_64-linux-gnu/gdk-pixbuf-2.0/2.10
.0/loaders/libpixbufloader-ani.so"
```

## 9. Usa grep -v para mostrar todas las líneas que NO contienen la palabra Linux.

```
jack@Ubuntu:~$ grep -rv "linux" .
./DAM1/redes/robinjack_dam_copia.txt:hola
./DAM1/robinjack_dam_26_09_2025.txt:hola
./DAM1/hola.sh:echo "Hola, soy Robin"
././bashrc:# ~/.bashrc: executed by bash(1) for non-login shells.
././bashrc:# see /usr/share/doc/bash/examples/startup-files (in the package b
ash-doc)
```

## 10. Usa grep -c para contar cuántas veces aparece la palabra Linux en un archivo.

>> grep -c cuenta en cuantas líneas aparece la palabra linux

```
jack@Ubuntu:~$ cat archivo.txt
linux hola linux
adios linux adios
jack@Ubuntu:~$ grep -c "linux" archivo.txt
2
```

## 11. Busca dentro de todos los archivos .conf del sistema la palabra network.

👉 Pista: `grep -r "network" /etc/*.conf`

```
jack@Ubuntu:~$ grep -r "network" /etc/*.conf
/etc/dhcpd.conf:# Respect the network MTU. This is applied to DHCP routes.
/etc/dhcpd.conf:# Request a hostname from the network
/etc/nsswitch.conf:networks:      files
/etc/rygel.conf:# List of network interfaces to attach rygel to. You can also use network IP or
```

### ♦ Ejercicios con locate

## 12. Instala mlocate y actualiza la base de datos con: `sudo updatedb`.

```
$ sudo apt-get install locate
```

```
$ sudo updatedb
```

## 13. Usa locate passwd para encontrar el archivo en el sistema.

```
$ locate passwd
```

## 14. Usa locate bin para ver cuántos directorios o archivos contienen esa palabra en su ruta.

```
$ locate bin
```

## 15. Busca con locate -i la palabra shadow (ignorando mayúsculas).

```
$ locate -i shadow
```

## 16. Busca archivos que terminen en .conf.

👉 Pista: `locate "*.conf"`

```
$ locate "*.conf"
```

Pregunta de reflexión: ¿Qué diferencia principal hay entre find y locate?

>> locate es mucho más rápido porque usa una base de datos indexada creada al usar updatedb, find es más lento pero más preciso si necesitas búsquedas con más condiciones.

## Parte 6: Procesos en Linux

### 1. Muestra la lista de procesos que se están ejecutando en tu sistema.

```
jack@Ubuntu:~$ ps
  PID TTY          TIME CMD
 2823 pts/0    00:00:00 bash
 3117 pts/0    00:00:00 ps
```

### 2. Identifica el PID (número de proceso) de tu terminal.

```
  PID TTY          TIME CMD
 2823 pts/0    00:00:00 bash
```

### 3. Abre una aplicación, localiza su proceso y finalízalo usando el comando para matar procesos.

```
jack      3210 64.6  4.7 3138664 386140 ?
```

```
Sl  16:59   0:18 /snap/firefox/6966/usr/lib/firefox/firefox
```

```
$ kill 3210
```

#### 4. Usa el comando gráfico xkill para cerrar una ventana haciendo clic sobre ella.

>> Esta distribución de ubuntu virtualizada no permite el funcionamiento correcto de xkill

Pregunta de reflexión: ¿Qué diferencia hay entre terminar un proceso desde la terminal y cerrarlo desde la interfaz gráfica?

>> Desde la terminal puedes usar diferentes comandos para cerrar una aplicación de distintas formas. Usando la interfaz gráfica tenemos menos control, únicamente nos llega un mensaje de confirmación.

### Parte 7: Introducción a Vim

#### 1. Instala Vim con: `sudo apt install vim`.

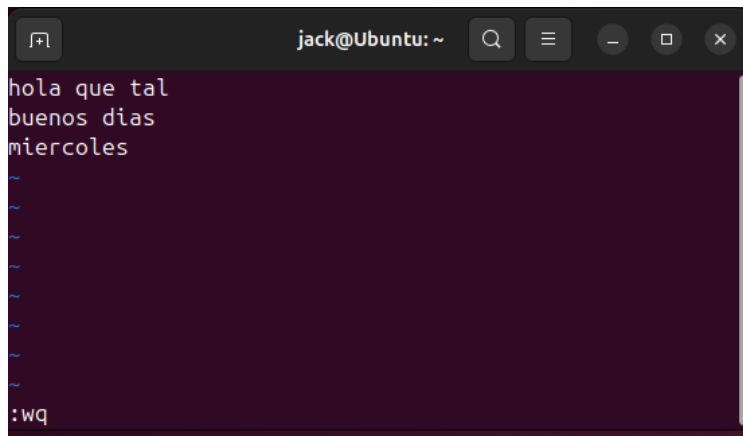
```
$ sudo apt-get install vim
```

#### 2. Abre un archivo llamado prueba.txt con Vim.

```
$ vim prueba.txt
```

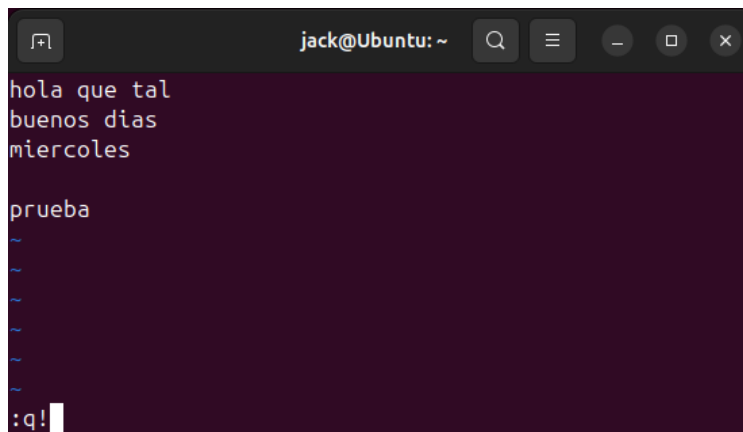
#### 3. Escribe un texto cualquiera (recuerda: pulsa i para entrar en modo inserción).

#### 4. Guarda los cambios y sal con ESC + :wq.



#### 5. Abre de nuevo el archivo y añade otra línea.

#### 6. Prueba a salir sin guardar usando ESC + :q!.



### 7. Busca una palabra dentro del archivo usando /palabra.

```
$ vim prueba.txt  
ESC  
/prueba
```

### 8. Elimina una línea completa con dd.

```
$ vim prueba.txt  
ESC  
dd
```

Pregunta de reflexión: ¿Qué ventajas tiene aprender a usar Vim en lugar de depender solo de editores gráficos?

>> Aprender muy bien Vim tiene sus ventajas porque es muy rápido, funciona en cualquier terminal y tiene atajos muy potentes, y puedes trabajar remotamente en servidores sin la necesidad de una GUI.

## Parte 8: Preguntas finales

### 1. ¿Qué diferencia hay entre un usuario normal y un usuario administrador con sudo?

Un usuario normal tiene permisos limitados por seguridad, un usuario con sudo puede ejecutar comandos como root, permitiendo realizar cualquier acción para administrar el sistema.

### 2. ¿Qué ventajas tiene organizar usuarios en grupos?

Con los grupos es más fácil gestionar permisos y aplicar acceso a ficheros, carpetas, etc. sin cambiar permisos usuario por usuario.

### 3. ¿Qué ventajas tiene buscar con grep frente a abrir manualmente cada archivo?

grep busca texto dentro de muchos archivos y encuentra coincidencias sin tener que abrir cada archivo uno por uno.

### 4. ¿Qué ventajas tiene locate frente a find?

locate es mucho más rápido porque usa una base de datos indexada creada al usar updatedb.

### 5. ¿Qué riesgos puede tener matar procesos del sistema?

Matar procesos puede causar pérdida de datos o incluso puedes bloquear el sistema si eliminas procesos esenciales.