BIG DATA

2025/26

Ciclo	Especialización IA & BIG DATA
Nombre	Rodrigo Medina
Correo	YMQ06518@educastur.es
Nº Unidad Didáctica	01

PR_01.3

1. Comandos de Información y Preparación

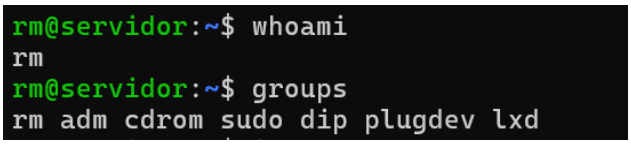
1. **Identidad del Usuario:** Abre una terminal y ejecuta un comando para saber qué usuario eres y a qué grupos perteneces.

   Usuario:

   ```
   whoami
   ```

   Grupos:

   ```
   groups
   ```



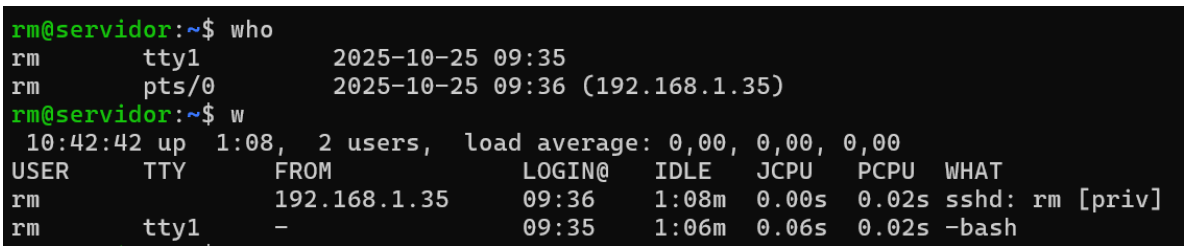
2. **Usuarios Conectados:** Muestra quién está conectado actualmente al sistema. Luego, ejecuta otro comando que te dé información más detallada, como el tiempo que llevan conectados y qué están ejecutando.

   Usuario conectado:

   ```
   who
   ```

   Información más detallada:

   ```
   w
   ```



3. **Historial de Conexiones**: Lista los últimos inicios de sesión en el sistema.

```
last
```

```
rm@servidor:~$ last
rm       pts/0        192.168.1.35     Sat Oct 25 09:36   still logged in
reboot   system boot  6.8.0-86-generic Sat Oct 25 09:34   still running
rm       pts/0        192.168.1.35     Fri Oct 24 18:31 - 18:34  (00:03)
reboot   system boot  6.8.0-86-generic Fri Oct 24 18:30   still running
rm       pts/0        192.168.1.35     Fri Oct 24 18:28 - 18:30  (00:02)
reboot   system boot  6.8.0-86-generic Fri Oct 24 18:22 - 18:30  (00:07)
rm       pts/0        192.168.1.35     Fri Oct 24 18:09 - 18:26  (00:17)
rm       pts/0        192.168.1.35     Fri Oct 24 17:59 - 17:59  (00:00)
reboot   system boot  6.8.0-86-generic Fri Oct 24 17:57 - 18:26  (00:28)
reboot   system boot  6.8.0-86-generic Fri Oct 24 17:56 - 17:58  (00:01)
rm       pts/0        192.168.1.35     Fri Oct 24 16:43 - 17:55  (01:12)
reboot   system boot  6.8.0-86-generic Fri Oct 24 16:41 - 17:58  (01:17)
rm       pts/0        192.168.1.35     Thu Oct 23 17:24 - crash  (23:17)
reboot   system boot  6.8.0-86-generic Thu Oct 23 17:21 - 17:58 (1+00:36)
```

4. **Crear Entorno de Trabajo:** En tu directorio personal (/home/tu_usuario), crea una carpeta principal para todos los ejercicios llamada practicas_linux .

```
mkdir practicas_linux
```

```
rm@servidor:~$ mkdir practicas_linux
rm@servidor:~$ ls -l
total 8
drwxrwxr-x 2 rm rm 4096 oct 25 10:53 practicas_linux
```

5. **Estructura de Directorios:** Dentro de practicas_linux , crea la siguiente estructura de directorios: proyectos , documentos y scripts .

```
mkdir practicas_linux/proyectos practicas_linux/documentos
practicas_linux/scripts
```

```
rm@servidor:~$ mkdir practicas_linux/proyectos practicas_linux/documentos practicas_linux/scripts
```

2. Gestión de Usuarios y Grupos

1. **Crear Grupos:** Crea tres nuevos grupos en el sistema: desarrolladores , y becarios .

```
sudo groupadd -g 1001 desarrolladores
sudo groupadd -g 1002 analistas
sudo groupadd -g 1003 becarios
```

```
rm@servidor:~$ sudo groupadd -g 1001 desarrolladores
```

```
rm@servidor:~$ sudo groupadd -g 1002 analistas
```

```
rm@servidor:~$ sudo groupadd -g 1003 becarios
```

2. **Verificar Grupos:** Confirma que los grupos se han creado correctamente buscando sus nombres en el archivo /etc/group .

```
cat /etc/group
```

```
desarrolladores:x:1001:
analistas:x:1002:
becarios:x:1003:
```

3. **Crear un Usuario Básico:** Crea un nuevo usuario llamado juan

```
sudo useradd juan
```

```
rm@servidor:~$ sudo useradd juan
```

Comprobar usuarios existentes:

```
cat /etc/passwd
```

```
rm:x:1000:1000:rm:/home/rm:/bin/bash
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
juan:x:1001:1004::/home/juan:/bin/sh
```

4. **Crear Usuario con Grupo Primario:** Crea una usuaria llamada ana asígnala directamente al grupo primario desarrolladores**.**

   -g → Grupo primario

```
sudo useradd -g desarrolladores ana
```

```
rm@servidor:~$ sudo useradd -g desarrolladores ana
ana:x:1002:1001::/home/ana:/bin/sh
```

5. **Crear Usuario Completo:** Crea un usuario ana y david asignándolo al grupo primario analistas y, a la vez, como miembro de los grupos secundarios desarrolladores y becarios **.**

   -G → grupos secundarios

```
sudo useradd -g analistas -G desarrolladores,becarios
david
```

```
rm@servidor:~$ sudo useradd -g analistas -G desarrolladores,becarios david
david:x:1003:1002::/home/david:/bin/sh
```

6. **Establecer Contraseñas:** Asigna una contraseña a los usuarios david .

```
sudo passwd usuario
```

```
rm@servidor:~$ sudo passwd juan
New password:
Retype new password:
passwd: password updated successfully
rm@servidor:~$ sudo passwd ana
New password:
Retype new password:
passwd: password updated successfully
rm@servidor:~$ sudo passwd david
New password:
Retype new password:
passwd: password updated successfully
```

7. **Verificar Usuarios:** Comprueba que los tres nuevos usuarios existen en el sistema, inspeccionando el final del archivo /etc/passwd .

```
cat /etc/passwd
```

```
juan:x:1001:1004::/home/juan:/bin/sh
ana:x:1002:1001::/home/ana:/bin/sh
david:x:1003:1002::/home/david:/bin/sh
```

8. **Cambiar de Usuario:** Conviértete en el usuario juan usando el comando su . Una vez dentro de su sesión, comprueba quién eres y en qué directorio te PR_01.3 1 encuentras. Vuelve a tu sesión de usuario original

```
su juan
whoami
pwd
```

```
rm@servidor:~$ su juan
Password:
$ whoami
juan
$ pwd
/home/rm
$ su rm
Password:
rm@servidor:~$
```

9. **Modificar Grupos de un Usuario:** Modifica al usuario juan para que su grupo primario sea becarios y añádelo también al grupo secundario analistas

```
sudo usermod -g becarios -G analistas juan
```

```
rm@servidor:~$ sudo usermod -g becarios -G analistas juan
```

10. **Verificar Modificación:** Comprueba que los cambios del usuario juan se han aplicado correctamente.

```
cat /etc/passwd | grep juan
```

```
rm@servidor:~$ cat /etc/passwd | grep juan
juan:x:1001:1003::/home/juan:/bin/sh
```

```
groups juan
```

```
rm@servidor:~$ groups juan
juan : becarios analistas
```

```
id juan
```

```
rm@servidor:~$ id juan
uid=1001(juan) gid=1003(becarios) groups=1003(becarios),1002(analistas)
```

11. **Bloquear una Cuenta:** Bloquea la cuenta del usuario juan para que no pueda iniciar sesión

-L → bloquear usuario (Locked)

```
sudo usermod -L juan
```

```
rm@servidor:~$ sudo usermod -L juan
```

12. **Intentar Cambiar a Usuario Bloqueado**: Intenta convertirte en el usuario juan de nuevo. Debería fallar.

```
su juan
```

```
rm@servidor:~$ su juan
Password:
su: Authentication failure
```

13. **Desbloquear una Cuenta:** Desbloquea la cuenta del usuario juan

    -U → Desbloquar (Unlocked)

```
sudo usermod -U juan
```

```
rm@servidor:~$ sudo usermod -U juan
```

14. **Eliminar un Grupo:** Elimina el grupo becarios . ¿Qué ocurre? Nota: Fallará si algún usuario lo tiene como grupo primario).

```
groupdel becarios
```

```
rm@servidor:~$ groupdel becarios
groupdel: cannot remove the primary group of user 'juan'
```

No se puede eliminar el grupo porque hay un usuario en ese grupo (juan)

15. **Eliminar Usuario y su Directorio:** Elimina al usuario juan y asegúrate de que su directorio personal (/home/juan) también se borre.

    -r → Eliminar directorio personal del usuario (Remove)

```
userdel -r juan
```

```
rm@servidor:~$ sudo userdel -r juan
```

```
rm@servidor:~$ id juan
id: ‘juan’: no such user
```

3. Permisos y Propiedad de Archivos

1. **Crear Archivos de Prueba:** Dentro de la carpeta vacío llamado informe.txt . Dentro de lanzar_app.sh . proyectos , crea un archivo scripts , crea otro archivo vacío llamado

```
rm@servidor:~$ cd practicas_linux/
rm@servidor:~/practicas_linux$ cd ..
rm@servidor:~$ cd practicas_linux/proyectos/
rm@servidor:~/practicas_linux/proyectos$ touch informe.txt
rm@servidor:~/practicas_linux/proyectos$ touch ../scripts/lanzar_app.sh
rm@servidor:~/practicas_linux/proyectos$
```

2. **Ver Permisos**: Muestra los permisos por defecto de los archivos y directorios que has creado. Anota quién es el propietario y el grupo.

```
rm@servidor:~$ ls -l practicas_linux/proyectos/informe.txt practicas_linux/scripts/lanzar_app.sh
-rw-rw-r-- 1 rm rm 0 oct 27 18:18 practicas_linux/proyectos/informe.txt
-rw-rw-r-- 1 rm rm 0 oct 27 18:19 practicas_linux/scripts/lanzar_app.sh
```

-rw-rw-r—

Propietario: Lectura y escritura

Grupo: Lectura y escritura

Otros; Lectura

En ambos archivos, el propietario es el usuario rm y el grupo rm

3. **Cambiar Propietario**: Cambia el propietario del archivo pertenezca a la usuaria ana .

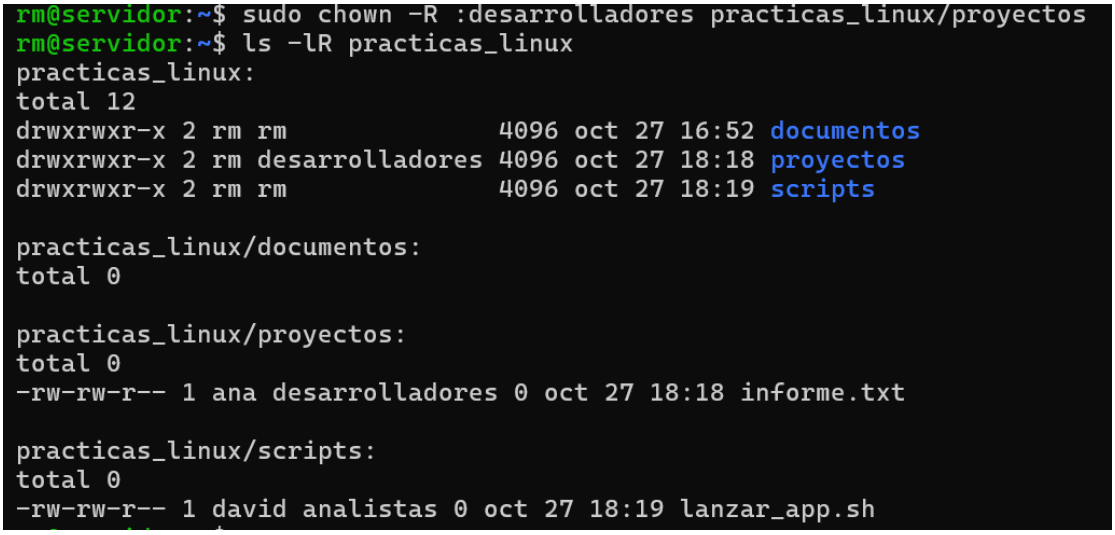
```
sudo chown ana practicas_linux/proyectos/informe.txt
```

```
rm@servidor:~$ sudo chown ana practicas_linux/proyectos/informe.txt
[sudo] password for rm:
rm@servidor:~$ ls -l practicas_linux/proyectos/informe.txt practicas_linux/scripts/lanzar_app.sh
-rw-rw-r-- 1 ana rm 0 oct 27 18:18 practicas_linux/proyectos/informe.txt
-rw-rw-r-- 1 rm  rm 0 oct 27 18:19 practicas_linux/scripts/lanzar_app.sh
```

4. **Cambiar Grupo:** Cambia el grupo del directorio proyectos para que pertenezca al grupo desarrolladores .

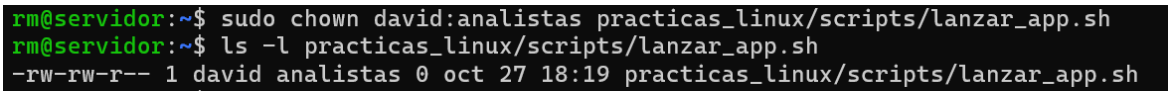
-R → de forma recursiva para cambiar también los permisos de los archivos y directorios que contiene

```
sudo chown -R :desarrolladores practicas_linux/proyectos
```



5. **Cambiar Propietario y Grupo**: Cambia el propietario y el grupo del archivo lanzar_app.sh para que pertenezcan al usuario respectivamente, con un solo comando. david y al grupo analistas ,

```
sudo chown david:analistas rutaArchivo
```



6. **Permisos con Notación Octal (Archivo):** Usa la notación numérica (octal) para asignar los siguientes permisos a informe.txt : el propietario (ana) puede leer y escribir; el grupo (desarrolladores) solo puede leer; y los otros no tienen ningún permiso.

   Propietario: Leer y escribir (4 +2) 6

   Grupo: Leer (4)

   Otros: Nada (0)

   640

```
sudo chmod 640 rutaArchivo
```

```
rm@servidor:~$ sudo chmod 640 practicas_linux/proyectos/informe.txt
rm@servidor:~$ ls -l practicas_linux/proyectos/
total 0
-rw-r----- 1 ana desarrolladores 0 oct 27 18:18 informe.txt
```

7. **Permisos con Notación Octal (Directorio**): Asigna permisos de lectura, escritura y ejecución para el propietario y solo de lectura y ejecución para los miembros del grupo al directorio documentos .

   Propietario: Leer , escribir y ejecutar (4 +2+1) 7

   Grupo: Leer y ejecutar (4+1) 5

   Otros: Nada (0)

   750

```
sudo chmod 750 rutaDirectorio
```

```
rm@servidor:~$ sudo chmod 750 practicas_linux/documentos/
rm@servidor:~$ ls -l practicas_linux/
total 12
drwxr-x--- 2 rm rm             4096 oct 27 16:52 documentos
drwxrwxr-x 2 rm desarrolladores 4096 oct 27 18:18 proyectos
drwxrwxr-x 2 rm rm             4096 oct 27 18:19 scripts
```

8. **Verificar Permisos:** Lista el contenido de practicas_linux para verificar que todos los cambios de propietario y permisos se han aplicado correctamente. □□ Permisos con Notación Simbólica □Añadir): Usa la notación simbólica para añadir el permiso de ejecución al propietario del script lanzar_app.sh .

-R → de forma recursiva para cambiar también los permisos de los archivos y directorios que contiene

```
ls -lR practicas_linux
```

```
rm@servidor:~$ ls -lR practicas_linux/
practicas_linux/:
total 12
drwxr-x--- 2 rm rm              4096 oct 27 16:52 documentos
drwxrwxr-x 2 rm desarrolladores 4096 oct 27 18:18 proyectos
drwxrwxr-x 2 rm rm              4096 oct 27 18:19 scripts

practicas_linux/documentos:
total 0

practicas_linux/proyectos:
total 0
-rw-r----- 1 ana desarrolladores 0 oct 27 18:18 informe.txt

practicas_linux/scripts:
total 0
-rw-rw-r-- 1 david analistas 0 oct 27 18:19 lanzar_app.sh
```

9. **Permisos con Notación Simbólica □Añadir):** Usa la notación simbólica para añadir el permiso de ejecución al propietario del script lanzar_app.sh **.**

```
sudo chmod u+x practicas_linux/scripts/lanzar_app.sh
```

```
rm@servidor:~$ sudo chmod u+x practicas_linux/scripts/lanzar_app.sh
```

```
rm@servidor:~$ ls -lR practicas_linux/scripts/
practicas_linux/scripts/:
total 0
-rwxrw-r-- 1 david analistas 0 oct 27 18:19 lanzar_app.sh
```

10. **Permisos con Notación Simbólica Quitar**): Quita el permiso de lectura al "resto del mundo" (otros) en el directorio proyectos .

```
sudo chmod o-r practicas_linux/proyectos
```

```
rm@servidor:~$ sudo chmod o-r practicas_linux/proyectos/
rm@servidor:~$ ls -lR practicas_linux/
practicas_linux/:
total 12
drwxr-x--- 2 rm rm             4096 oct 27 16:52 documentos
drwxrwx--x 2 rm desarrolladores 4096 oct 27 18:18 proyectos
drwxrwxr-x 2 rm rm             4096 oct 27 18:19 scripts
```

11. **Permisos Recursivos:** Dentro de proyectos , crea una nueva carpeta version2 con un archivo notas.txt dentro. Luego, cambia el propietario de la carpeta proyectos y todo su contenido para que pertenezca a david con un solo comando recursivo.

```
sudo chown -R david practicas_linux/proyectos
```

```
rm@servidor:~$ sudo chown david -R practicas_linux/proyectos/
```

12. **Permiso Especial SGID en Directorio:** Establece el permiso especial SGID en el directorio documentos . Después, cambia a ser el usuario david (su david) y crea un nuevo archivo dentro de documentos . Verifica a qué grupo pertenece el nuevo archivo (debería heredar el del directorio documentos). Vuelve a tu usuario.

SGID → g+s

```
sudo chmod g+s practicas_linux/documentos
```

```
rm@servidor:~$ sudo chmod g+s practicas_linux/documentos
```

13. **Permiso Especial SUID**□ Establece el permiso SUID en el script lanzar_app.sh . □Nota: Explica a tus alumnos qué implicaría esto si fuera un programa compilado).

SUID → u+s

```
sudo chmod u+s rutaArchivo
```

```
rm@servidor:~$ sudo chmod u+s practicas_linux/scripts/lanzar_app.sh
rm@servidor:~$ ls -lR practicas_linux/scripts/
practicas_linux/scripts/:
total 0
-rwsrw-r-- 1 david analistas 0 oct 27 18:19 lanzar_app.sh
```

14. **Comprobar umask :** Muestra el valor umask actual de tu sesión.

```
umask
```

```
rm@servidor:~$ umask
0002
```

15. **Efecto de umask :** Cambia temporalmente tu umask a archivo llamado 077 . Crea un nuevo privado.txt . Comprueba sus permisos por defecto. Luego, restaura el umask a su valor original.

```
umask 077
```

```
rm@servidor:~$ umask 077
rm@servidor:~$ touch privado077.txt
rm@servidor:~$ umask 0002
rm@servidor:~$ touch privado0002.txt
rm@servidor:~$ ls -l privado077.txt privado0002.txt
-rw-rw-r-- 1 rm rm 0 oct 28 18:12 privado0002.txt
-rw------- 1 rm rm 0 oct 28 18:12 privado077.txt
rm@servidor:~$ umask 0002
rm@servidor:~$ umask
0002
```

4. Gestión de Servicios con systemctl

1. **Estado Detallado de un Servicio**: Comprueba el estado completo del servicio cups . Analiza la salida: ¿está activo (active), cargado (loaded) y habilitado (enabled □? Anota las últimas líneas de su registro (log) que aparecen.

```
sudo systemctl status cups
```

Unit cups.service could not be found.

```
rm@servidor:~$ sudo systemctl status cups
Unit cups.service could not be found.
```

```
sudo apt install cups
```

```
rm@servidor:~$ sudo systemctl status cups
● cups.service - CUPS Scheduler
     Loaded: loaded (/usr/lib/systemd/system/cups.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-10-28 18:40:06 UTC; 3min 28s ago
TriggeredBy: ● cups.path
             ● cups.socket
       Docs: man:cupsd(8)
   Main PID: 3557 (cupsd)
     Status: "Scheduler is running..."
      Tasks: 1 (limit: 9433)
     Memory: 2.7M (peak: 17.2M)
        CPU: 658ms
     CGroup: /system.slice/cups.service
             └─3557 /usr/sbin/cupsd -l

oct 28 18:40:06 servidor systemd[1]: Starting cups.service - CUPS Scheduler...
```

2. **Comprobación Rápida:** Utiliza un comando más directo para verificar si el servicio cups está actualmente en ejecución (activo). La salida de este comando debería ser simplemente active o inactive .

```
sudo systemctl is-active cups
```

```
rm@servidor:~$ sudo systemctl is-active cups
active
```

3. **Ver Archivo de Unidad:** Muestra el contenido del archivo de unidad del servicio cups (cups.service). Esto te permitirá ver cómo está definido el servicio.

```
sudo systemctl cat cups
```

```
rm@servidor:~$ sudo systemctl cat cups
# /usr/lib/systemd/system/cups.service
[Unit]
Description=CUPS Scheduler
Documentation=man:cupsd(8)
After=network.target nss-user-lookup.target nslcd.service
Requires=cups.socket

[Service]
ExecStart=/usr/sbin/cupsd -l
Type=notify
Restart=on-failure


[Install]
Also=cups.socket cups.path
WantedBy=printer.target multi-user.target
```

4. **Detener un Servicio:** Detén la ejecución del servicio estado de nuevo para confirmar que está cups . Comprueba su inactive (dead) .

```
sudo systemctl stop cups
```

```
rm@servidor:~$ sudo systemctl stop cups
rm@servidor:~$ sudo systemctl is-active cups
inactive
```

5. **Iniciar un Servicio:** Vuelve a iniciar el servicio cups . Verifica una vez más que ha vuelto al estado active (running) .

```
sudo systemctl start cups
```

```
rm@servidor:~$ sudo systemctl start cups
rm@servidor:~$ sudo systemctl is-active cups
active
```

6. **Reiniciar un Servicio:** El comando restart es muy común tras un cambio de configuración. Ejecútalo para el servicio cups .

```
sudo systemctl restart cups
```

```
rm@servidor:~$ sudo systemctl restart cups
rm@servidor:~$ sudo systemctl is-active cups
active
```

7. **Habilitar para el Arranque**: Asegúrate de que el servicio cups esté configurado para iniciarse automáticamente cada vez que el sistema arranque.

```
sudo systemctl enable cups
```

```
rm@servidor:~$ sudo systemctl enable cups
Synchronizing state of cups.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable cups
```

8. **Verificar si está Habilitado:** Usa un comando específico para preguntar si cups está habilitado. La salida debería ser enabled o disabled .

```
sudo systemctl status cups
```

```
rm@servidor:~$ sudo systemctl status cups
● cups.service - CUPS Scheduler
     Loaded: loaded (/usr/lib/systemd/system/cups.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-10-28 18:49:09 UTC; 1min 47s ago
TriggeredBy: ○ cups.path
             ● cups.socket
       Docs: man:cupsd(8)
   Main PID: 3844 (cupsd)
     Status: "Scheduler is running..."
      Tasks: 1 (limit: 9433)
     Memory: 1.7M (peak: 1.8M)
        CPU: 14ms
     CGroup: /system.slice/cups.service
             └─3844 /usr/sbin/cupsd -l

oct 28 18:49:09 servidor systemd[1]: Starting cups.service - CUPS Scheduler...
oct 28 18:49:09 servidor systemd[1]: Started cups.service - CUPS Scheduler.
```

9. **Deshabilitar para el Arranque:** Ahora, desactiva el servicio no se inicie automáticamente. cups para que

```
sudo systemctl disable cups
```

```
rm@servidor:~$ sudo systemctl disable cups
Synchronizing state of cups.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install disable cups
Removed "/etc/systemd/system/printer.target.wants/cups.service".
Removed "/etc/systemd/system/sockets.target.wants/cups.socket".
Removed "/etc/systemd/system/multi-user.target.wants/cups.path".
Removed "/etc/systemd/system/multi-user.target.wants/cups.service".
Disabling 'cups.service', but its triggering units are still active:
cups.socket
```

```
rm@servidor:~$ sudo systemctl status cups
● cups.service - CUPS Scheduler
     Loaded: loaded (/usr/lib/systemd/system/cups.service; disabled; preset: enabled)
     Active: active (running) since Tue 2025-10-28 18:49:09 UTC; 3min 6s ago
```

10. **Enmascarar un Servicio:** El enmascaramiento es una forma más contundente de deshabilitar, ya que impide cualquier tipo de inicio (manual o automático). Enmascara el servicio cups . Intenta iniciarlo después. Debería fallar. No olvides desenmascararlo (unmask) al terminar el ejercicio.

```
sudo systemctl mask cups
sudo systemctl start cups
sudo systemctl unmask cups
```

```
rm@servidor:~$ sudo systemctl mask cups
Created symlink /etc/systemd/system/cups.service → /dev/null.
Masking 'cups.service', but its triggering units are still active:
cups.socket
```

```
rm@servidor:~$ sudo systemctl start cups
Failed to start cups.service: Unit cups.service is masked.
```

```
rm@servidor:~$ sudo systemctl unmask cups
Removed "/etc/systemd/system/cups.service".
```

5. Gestión de ufw

1. **Comprobar Estado y Activar UFW** * Primero, ejecuta un comando para verificar el estado actual del firewall. Probablemente estará inactivo. * A continuación, activa UFW. Presta atención al mensaje de advertencia, especialmente si estás conectado por SSH

```
sudo ufw status
```

```
rm@servidor:~$ sudo ufw status
Status: inactive
```

```
sudo ufw enable
```

```
rm@servidor:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

2. **Permitir un Servicio Web HTTP** * Imagina que tu servidor necesita alojar una página web. Añade una regla para permitir todas las conexiones entrantes para el servicio http . * Verifica el estado del firewall de nuevo para confirmar que la regla (y el puerto 80 se ha añadido correctamente.

```
sudo ufw allow http
```

```
rm@servidor:~$ sudo ufw allow http
Rule added
Rule added (v6)
```

```
sudo ufw status
```

```
rm@servidor:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
80/tcp (v6)                ALLOW       Anywhere (v6)
```

3. **Abrir un Puerto Específico:** * Imagina que estás ejecutando un servidor de aplicaciones web en el puerto 8080. Añade una regla para permitir las conexiones entrantes ese puerto. TCP a □□

```
sudo ufw allow 8080/tcp
```

```
rm@servidor:~$ sudo ufw allow 8080/tcp
Rule added
Rule added (v6)
rm@servidor:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
8080/tcp                   ALLOW       Anywhere
80/tcp (v6)                ALLOW       Anywhere (v6)
8080/tcp (v6)              ALLOW       Anywhere (v6)
```

4. **Permitir un Rango de Puertos:** * Supón que una aplicación FTP necesita un rango de puertos pasivos. Añade una regla para permitir las conexiones desde el 3000 al 3100. TCP en el rango de puertos

```
sudo ufw allow 3000:3100/tcp
```

```
rm@servidor:~$ sudo ufw allow 3000:3100/tcp
Rule added
Rule added (v6)
rm@servidor:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
8080/tcp                   ALLOW       Anywhere
3000:3100/tcp              ALLOW       Anywhere
80/tcp (v6)                ALLOW       Anywhere (v6)
8080/tcp (v6)              ALLOW       Anywhere (v6)
3000:3100/tcp (v6)         ALLOW       Anywhere (v6)
```

5. **Bloquear una Dirección IP** * Por seguridad, has detectado actividad sospechosa desde la IP 192.168.100.50 . Añade una regla para denegar todas las conexiones provenientes de esa dirección IP.

```
sudo ufw deny from 192.168.100.50
```

```
rm@servidor:~$ sudo ufw deny from 192.168.100.50
Rule added
rm@servidor:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
8080/tcp                   ALLOW       Anywhere
3000:3100/tcp              ALLOW       Anywhere
Anywhere                   DENY        192.168.100.50
80/tcp (v6)                ALLOW       Anywhere (v6)
8080/tcp (v6)              ALLOW       Anywhere (v6)
3000:3100/tcp (v6)         ALLOW       Anywhere (v6)
```

6. **Listar Reglas para Borrar:** * Muestra todas las reglas activas del firewall, pero esta vez de forma numerada, para prepararte para eliminar una de ellas.

```
sudo ufw status numbered
```

```
rm@servidor:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 80/tcp                     ALLOW IN    Anywhere
[ 2] 8080/tcp                   ALLOW IN    Anywhere
[ 3] 3000:3100/tcp              ALLOW IN    Anywhere
[ 4] Anywhere                   DENY IN     192.168.100.50
[ 5] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 6] 8080/tcp (v6)              ALLOW IN    Anywhere (v6)
[ 7] 3000:3100/tcp (v6)         ALLOW IN    Anywhere (v6)
```

7. **Eliminar una Regla:** * Basándote en la lista del ejercicio anterior, elimina la regla que creaste para el puerto 8080 . * Vuelve a listar las reglas (de forma normal o numerada) para confirmar que la regla ha sido eliminada correctamente.

*Hay que borrar 2 reglas para el puerto 8080, para ipv4 e ipv6

```
sudo ufw delete 2
sudo ufw delete 6
```

```
rm@servidor:~$ sudo ufw delete 2
Deleting:
 allow 8080/tcp
Proceed with operation (y|n)? y
Rule deleted
```

```
rm@servidor:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 80/tcp                     ALLOW IN    Anywhere
[ 2] 3000:3100/tcp              ALLOW IN    Anywhere
[ 3] Anywhere                   DENY IN     192.168.100.50
[ 4] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 5] 3000:3100/tcp (v6)         ALLOW IN    Anywhere (v6)
```