PR_01.3

Bloque 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.

```
brayan1@brayans:~$ whoami
brayan1
brayan1@brayans:~$ groups
brayan1 adm cdrom sudo dip plugdev lxd
brayan1@brayans:~$
```

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.
   who o w

```
brayan1@brayans:~$ w
 20:32:24 up 22 min,  2 users,  load average: 0,00, 0,00, 0,00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU  WHAT
brayan1           192.168.1.139    20:29   22:42   0.00s  0.03s sshd: braya
brayan1  tty1     -                20:27    4:48   0.06s  0.03s -bash
brayan1@brayans:~$ who
brayan1  tty1         2025-10-29 20:27
brayan1  pts/0        2025-10-29 20:29 (192.168.1.139)
brayan1@brayans:~$
```

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

```
brayan1@brayans:~$ last
brayan1  pts/0        192.168.1.139    Wed Oct 29 20:29   still logged in
reboot   system boot  6.8.0-86-generic Wed Oct 29 20:09   still running
reboot   system boot  6.8.0-86-generic Wed Oct 29 19:58 - 19:58  (00:00)
brayan1  pts/0        192.168.1.139    Tue Oct 28 18:47 - 18:47  (00:00)
reboot   system boot  6.8.0-86-generic Tue Oct 28 18:44 - 18:47  (00:03)
reboot   system boot  6.8.0-86-generic Tue Oct 28 18:37 - 18:43  (00:06)

wtmp begins Tue Oct 28 18:37:15 2025
brayan1@brayans:~$
```

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

```
brayan1@brayans:~$ mkdir practicas_linux
brayan1@brayans:~$ ls
practicas_linux
brayan1@brayans:~$ pwd
/home/brayan1
brayan1@brayans:~$
```

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

```
brayan1@brayans:~$ cd practicas_linux
brayan1@brayans:~/practicas_linux$ mkdir proyectos documentos scripts
brayan1@brayans:~/practicas_linux$ ls
documentos  proyectos  scripts
brayan1@brayans:~/practicas_linux$
```

Bloque 2: Gestión de Usuarios y Grupos

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

```
brayan1@brayans:/$ sudo groupadd desarrolladores
groupadd: group 'desarrolladores' already exists
brayan1@brayans:/$ sudo groupadd analistas
groupadd: group 'analistas' already exists
brayan1@brayans:/$ sudo groupadd becarios
groupadd: group 'becarios' already exists
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ grep 'desarrolladores\|analistas\|becarios' /etc/group
desarrolladores:x:1001:
analistas:x:1002:
becarios:x:1003:
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo useradd juan
```

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

```
brayan1@brayans:/$ sudo useradd -g desarrolladores ana
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo useradd -g analistas -G desarrolladores,becarios dav
id
brayan1@brayans:/$
```

6. Establecer Contraseñas: Asigna una contraseña a los usuarios juan , ana y david .

```
brayan1@brayans:/$ sudo passwd juan
New password:
Retype new password:
passwd: password updated successfully
brayan1@brayans:/$ sudo passwd ana
New password:
Retype new password:
passwd: password updated successfully
brayan1@brayans:/$ sudo passwd david
New password:
Retype new password:
passwd: password updated successfully
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ tail -n 3 /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
brayan1@brayans:/$
```

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 encuentras. Vuelve a tu sesión de usuario original.

```
brayan1@brayans:/$ su juan
Password:
$ whoami
juan
$ pwd
/
$ exit
brayan1@brayans:/$ whoami
brayan1
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo usermod -g becarios -G analistas juan
brayan1@brayans:/$ id juan
```

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

```
uid=1001(juan) gid=1003(becarios) groups=1003(becarios),1002(analistas)
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo usermod -L juan
brayan1@brayans:/$ su juan
Password:
su: Authentication failure
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo usermod -L juan
brayan1@brayans:/$ su juan
Password:
su: Authentication failure
brayan1@brayans:/$
```

13. Desbloquear una Cuenta: Desbloquea la cuenta del usuario

```
brayan1@brayans:/$ sudo usermod -U juan
brayan1@brayans:/$ su juan
Password:
$ pwd
/
$ whoami
juan
$
```

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

```
brayan1@brayans:/$ sudo groupdel becarios
groupdel: cannot remove the primary group of user 'juan'
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo userdel -r juan
userdel: group juan not removed because it is not the primary group of user
juan.
userdel: juan mail spool (/var/mail/juan) not found
userdel: juan home directory (/home/juan) not found
brayan1@brayans:/$
```

Bloque 3: Permisos y Propiedad de Archivos

Realiza los siguientes ejercicios dentro de la carpeta practicas_linux . de tu directorio home

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

```
brayan1@brayans:/$ cd ~/practicas_linux/proyectos
brayan1@brayans:~/practicas_linux/proyectos$ touch informe.txt
brayan1@brayans:~/practicas_linux/proyectos$ cd ../scripts
brayan1@brayans:~/practicas_linux/scripts$ touch lanzar_app.sh
brayan1@brayans:~/practicas_linux/scripts$ ls -l
total 0
-rw-rw-r-- 1 brayan1 brayan1 0 oct 29 21:14 lanzar_app.sh
brayan1@brayans:~/practicas_linux/scripts$
```

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

```
brayan1@brayans:~/practicas_linux/scripts$ cd /
brayan1@brayans:/$ ls -l ~/practicas_linux
total 12
drwxrwxr-x 2 brayan1 brayan1 4096 oct 29 20:49 documentos
drwxrwxr-x 2 brayan1 brayan1 4096 oct 29 21:13 proyectos
drwxrwxr-x 2 brayan1 brayan1 4096 oct 29 21:14 scripts
brayan1@brayans:/$ ls -l ~/practicas_linux/proyectos
total 0
-rw-rw-r-- 1 brayan1 brayan1 0 oct 29 21:13 informe.txt
brayan1@brayans:/$ ls -l ~/practicas_linux/scripts
total 0
-rw-rw-r-- 1 brayan1 brayan1 0 oct 29 21:14 lanzar_app.sh
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo chown ana ~/practicas_linux/proyectos/informe.txt
brayan1@brayans:/$ ls -l ~/practicas_linux/proyectos/informe.txt
-rw-rw-r-- 1 ana brayan1 0 oct 29 21:13 /home/brayan1/practicas_linux/proyec
tos/informe.txt
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo chgrp desarrolladores ~/practicas_linux/proyectos
brayan1@brayans:/$ ls -l desarrolladores ~/practicas_linux/proyectos
ls: cannot access 'desarrolladores': No such file or directory
/home/brayan1/practicas_linux/proyectos:
total 0
-rw-rw-r-- 1 ana brayan1 0 oct 29 21:13 informe.txt
brayan1@brayans:/$
```

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 ,

```
brayan1@brayans:/$ sudo chown david:analistas ~/practicas_linux/scripts/lanz
ar_app.sh
```

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.

```
brayan1@brayans:/$ sudo chmod 640 ~/practicas_linux/proyectos/informe.txt
```

```
brayan1@brayans:/$ ls -l ~/practicas_linux/proyectos/informe.txt
-rw-r----- 1 ana brayan1 0 oct 29 21:13 /home/brayan1/practicas_linux/proyec
tos/informe.txt
brayan1@brayans:/$
```

Propietario RW, grupo R, otros sin permisos

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 .

```
brayan1@brayans:/$ sudo chmod 750 ~/practicas_linux/documentos
brayan1@brayans:/$ ls -l ~/practicas_linux/documentos
total 0
brayan1@brayans:/$
```

8. Verificar Permisos: Lista el contenido de practicas_linux para verificar que todos los cambios de propietario y permisos se han aplicado correctamente.

```
brayan1@brayans:/$ ls -l ~/practicas_linux
total 12
drwxr-x--- 2 brayan1 brayan1         4096 oct 29 20:49 documentos
drwxrwxr-x 2 brayan1 desarrolladores 4096 oct 29 21:13 proyectos
drwxrwxr-x 2 brayan1 brayan1         4096 oct 29 21:14 scripts
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ chmod u+x ~/practicas_linux/scripts/lanzar_app.sh
chmod: changing permissions of '/home/brayan1/practicas_linux/scripts/lanzar
_app.sh': Operation not permitted
brayan1@brayans:/$ sudo chmod u+x ~/practicas_linux/scripts/lanzar_app.sh
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ chmod o-r ~/practicas_linux/proyectos
brayan1@brayans:/$ ld -l ~/practicas_linux/proyectos
Command 'ld' not found, but can be installed with:
sudo apt install binutils
brayan1@brayans:/$ ls -l ~/practicas_linux/proyectos
total 0
-rw-r----- 1 ana brayan1 0 oct 29 21:13 informe.txt
brayan1@brayans:/$
```

11. Permisos Recursivos: Dentro de proyectos , crea una nueva carpeta con un archivo version2 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.

```
brayan1@brayans:/$ mkdir ~/practicas_linux/proyectos/version2
brayan1@brayans:/$ touch ~/practicas_linux/proyectos/version2/notas.txt
brayan1@brayans:/$ sudo chown -R david ~/practicas_linux/proyectos
brayan1@brayans:/$
```

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.

```
brayan1@brayans:/$ sudo chmod g+s ~/practicas_linux/documentos
brayan1@brayans:/$ su david
Password:
çwhoami
david
$ touch ~/practicas_linux/documentos/prueba.txt
touch: cannot touch '/home/david/practicas_linux/documentos/prueba.txt': No
such file or directory
$ ls -l ~/practicas_linux/documentos
ls: cannot access '/home/david/practicas_linux/documentos': No such file or
directory
$ exit
brayan1@brayans:/$
```

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).

```
brayan1@brayans:/$ sudo chmod u+s ~/practicas_linux/scripts/lanzar_app.sh
brayan1@brayans:/$
```

Si fuese un programa compliado se abriría

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

```
brayan1@brayans:/$ umask
0002
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ umask 077
brayan1@brayans:/$ touch ~/practicas_linux/privado.txt
brayan1@brayans:/$ ls -l ~/practicas_linux/privado.txt
-rw------- 1 brayan1 brayan1 0 oct 29 21:29 /home/brayan1/practicas_linux/pr
ivado.txt
brayan1@brayans:/$ umask 022
brayan1@brayans:/$
```

Bloque 4: Gestión de Servicios con systemctl

Nota: Para estos ejercicios, es seguro usar un servicio como cron / cups (impresión) o crond (tareas programadas). Evita usar servicios críticos como sshd si no estás seguro.

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.

```
brayan1@brayans:/$ systemctl status cups
Unit cups.service could not be found.
```

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 .

```
brayan1@brayans:/$ systemctl is-active cups
inactive
brayan1@brayans:/$
```

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.

```
brayan1@brayans:/$ cat /lib/systemd/system/cups.service
cat: /lib/systemd/system/cups.service: No such file or directory
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo systemctl stop cups
Failed to stop cups.service: Unit cups.service not loaded.
brayan1@brayans:/$ systemctl is-active cups
inactive
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo systemctl start cups
Failed to start cups.service: Unit cups.service not found.
brayan1@brayans:/$ systemctl is-active cups
inactive
brayan1@brayans:/$
```

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

```
Failed to restart cups.service: Unit cups.service not found.
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo systemctl enable cups
Failed to enable unit: Unit file cups.service does not exist.
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ systemctl is-enabled cups
not-found
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo systemctl disable cups
Failed to disable unit: Unit file cups.service does not exist.
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo systemctl mask cups
Unit cups.service does not exist, proceeding anyway.
Created symlink /etc/systemd/system/cups.service → /dev/null.
brayan1@brayans:/$ sudo systemctl start cups
Failed to start cups.service: Unit cups.service is masked.
brayan1@brayans:/$ sudo systemctl unmask cups
Removed "/etc/systemd/system/cups.service".
brayan1@brayans:/$
```

Bloque 4: 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.

```
brayan1@brayans:/$ sudo ufw status
Status: inactive
brayan1@brayans:/$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)?
y
Firewall is active and enabled on system startup
brayan1@brayans:/$
```

2. Permitir un Servicio We8b 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.

```
brayan1@brayans:/$ sudo ufw allow http
Rule added
Rule added (v6)
brayan1@brayans:/$ sudo ufw status
Status: active

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

brayan1@brayans:/$
```

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 TCP a ese puerto.

```
brayan1@brayans:/$ sudo ufw allow 8080/tcp
Rule added
Rule added (v6)
brayan1@brayans:/$
```

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

```
brayan1@brayans:/$ sudo ufw allow 3000:3100/tcp
Rule added
Rule added (v6)
brayan1@brayans:/$
```

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.

```
brayan1@brayans:/$ sudo ufw deny from 192.168.100.50
Rule added
brayan1@brayans:/$
```

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.

```
brayan1@brayans:/$ 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)

brayan1@brayans:/$
```

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