

# Gestión de Usuarios y Permisos en Linux

Este documento proporciona una guía práctica para la gestión de usuarios y permisos en sistemas Linux. Está dirigido a principiantes y usuarios intermedios que deseen aprender cómo administrar cuentas de usuario, asignar permisos a archivos y comprender los conceptos clave relacionados con la seguridad y administración del sistema.

## Esta guía ofrece:

- ❖ Creación, modificación y eliminación de usuarios y grupos.
- ❖ Asignación de permisos a archivos a través de métodos numéricos y alfabéticos.
- ❖ Cambio de contraseñas y administración de accesos en Linux.

## Sobre el Autor:

- Aidee Miguelina Lorenzo Mejía
- Estudiante de Seguridad Informática. Instituto Tecnológico de Las Américas.
- Creando contenido educativo para ayudar a otros a entender mejor Linux y la ciberseguridad.

# Contenido



## *Gestión de usuarios y permisos en Linux*

<b>Introducción.....</b>	<b>3</b>
¿Qué es Linux? .....	3
 <b>Usuarios y Grupos en Linux .....</b>	<b>4</b>
¿Qué es un usuario? .....	4
¿Qué es un grupo? .....	4
 <b>Usuarios en Linux .....</b>	<b>5</b>
Crear un usuario nuevo.....	5
Establecer y cambiar una contraseña.....	6
Verificar la existencia de un usuario .....	7
Cambio de usuario .....	7
Eliminación de un usuario .....	8
 <b>Grupos en Linux.....</b>	<b>9</b>
Creación de grupos .....	9
Asignación de usuarios a un grupo.....	10
Verificar los grupos de un usuario.....	11
Eliminación de un grupo.....	12
Eliminar un usuario de un grupo .....	13

<b>Permisos de Archivos en Linux .....</b>	<b>15</b>
¿Qué son los permisos en Linux? .....	15
Tipos de permisos .....	15
Visualización de permisos en archivos.....	16
Asignación de permisos en forma numérica.....	18
Asignación de permisos en forma alfabética .....	19
 <b>Conclusión y reflexión final.....</b>	<b>20</b>
Resumen de lo aprendido y reflexión personal .....	20
Importancia del tema .....	21

## Introducción

### ¿Qué es Linux?

Linux es un sistema operativo multiusuario, lo que significa que varias personas pueden acceder y operar en el mismo sistema al mismo tiempo. Para mantener la seguridad y organización, es fundamental gestionar correctamente los usuarios y los permisos de los archivos. Esta guía muestra cómo crear usuarios, asignarles permisos y entender la diferencia entre los distintos niveles de acceso.

## Usuarios y Grupos en Linux

### ¿Qué es un usuario?

En Linux, un usuario es una identidad que puede iniciar sesión y ejecutar procesos en el sistema. Cada usuario tiene un entorno, configuraciones y permisos.

### ¿Qué es un grupo?

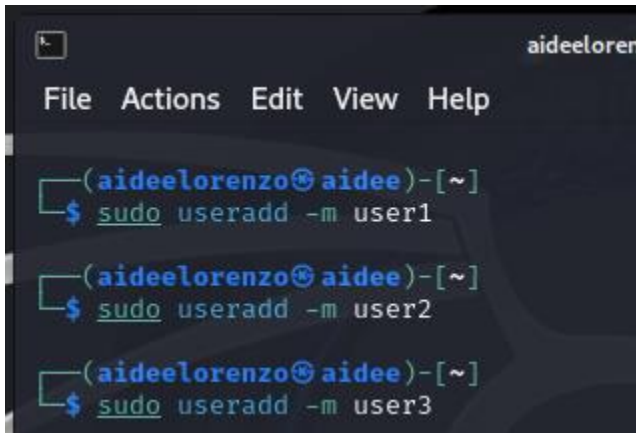
Un grupo en Linux es una colección de usuarios que comparten ciertos permisos y privilegios. Los grupos facilitan la administración de permisos de archivos y directorios al asignar accesos a múltiples usuarios simultáneamente.

## Usuarios en Linux

### Crear un usuario nuevo

Para agregar un nuevo usuario en Linux, usamos el comando: **useradd + <nombre\_usuario>**.

Ejemplo gráfico:



```
File Actions Edit View Help
(aideelorenzo@aidee)-[~]
$ sudo useradd -m user1
(aideelorenzo@aidee)-[~]
$ sudo useradd -m user2
(aideelorenzo@aidee)-[~]
$ sudo useradd -m user3
```

- **sudo:** ejecuta el comando con privilegios de administrador
- **useradd:** crea el usuario en el sistema
- **-m:** crea de forma automática un directorio personal (**/home/user1**) para el usuario.

## Establecer y cambiar una contraseña

Para mayor seguridad un usuario necesita contraseña para su identidad. Para establecer una contraseña usamos: `passwd + <nombre_usuario>`.

Se pedirá ingresar y confirmar la contraseña.

```
(aideelorenzo@aidee)-[~]  
$ sudo passwd user1  
New password:  
Retype new password:  
passwd: password updated successfully  
  
(aideelorenzo@aidee)-[~]  
$ sudo passwd user2  
New password:  
Retype new password:  
passwd: password updated successfully
```

De igual forma para cambiar la contraseña de un usuario existente, usamos el mismo comando. El sistema nos pedirá una nueva contraseña para el usuario.

```
(aideelorenzo@aidee)-[~]  
$ sudo passwd user2  
New password:  
Retype new password:  
passwd: password updated successfully
```

## Verificar la existencia de un usuario

Con el comando `cat /etc/passwd | grep <nombre_usuario>` se mostrará la información del usuario recién creado.

```
(aideelorenzo@aidee)-[~]
$ cat /etc/passwd | grep user1
user1:x:1001:1001::/home/user1:/bin/sh

(aideelorenzo@aidee)-[~]
$ cat /etc/passwd | grep user2
user2:x:1002:1002::/home/user2:/bin/sh

(aideelorenzo@aidee)-[~]
$ cat /etc/passwd | grep user3
user3:x:1003:1003::/home/user3:/bin/sh

(aideelorenzo@aidee)-[~]
$
```

## Cambio de usuario

Para cambiar a otro usuario sin cerrar sesión, en la terminal escribe: `su - <nombre_usuario>`

Para regresar al usuario anterior, usar: `exit`

```
(aideelorenzo@aidee)-[~]
$ su - user1
Password:
$
$ pwd
/home/user1
$
$
```

- **su** = switch user
- **- user1**: especifica a cuál usuario queremos cambiar.



```
(aideelorenzo@aidee)-[~]  
$ su - user3  
Password:  
$  
$ whoami  
user3  
$ pwd  
/home/user3  
$
```

## Eliminación de un usuario

Para eliminar un usuario, utilizar el comando: **userdel <nombre\_usuario>**

```
(aideelorenzo@aidee)-[~]  
$ sudo userdel user1  
  
(aideelorenzo@aidee)-[~]  
$ cat /etc/passwd | grep user1  
  
(aideelorenzo@aidee)-[~]  
$ cat /etc/passwd | grep /home  
aideelorenzo:x:1000:1000:aideelorenzo,,,:/home/aideelorenzo:/usr/bin/zsh  
user2:x:1002:1002::/home/user2:/bin/sh  
user3:x:1003:1003::/home/user3:/bin/sh
```

# Grupos en Linux

## Creación de grupos

Con ayuda del comando: **groupadd <nombre\_grupo>**, creamos un grupo.

```
(aideelorenzo@aidee)-[~]  
$ sudo groupadd grupo1  
  
(aideelorenzo@aidee)-[~]  
$ sudo groupadd grupo2  
  
(aideelorenzo@aidee)-[~]  
$ sudo groupadd grupo3  
  
(aideelorenzo@aidee)-[~]  
$ sudo groupadd grupo4 && sudo groupadd grupo5  
  
(aideelorenzo@aidee)-[~]  
$
```

**NOTA:** “&&” se ha utilizado para encadenar dos comandos. Si el primero se ejecuta de forma correcta, el segundo procederá a ejecutarse también.

## Asignación de usuarios a un grupo

El comando **usermod -aG <nombre\_grupo> <nombre\_usuario>**, hará que el usuario mencionado sea asignado a un grupo específico.

Un usuario puede estar en dos o más grupos a la vez. Esto siempre dependerá de la demanda de responsabilidades que tenga su rol.



```
(aideelorenzo@aidee)-[~]  
$ sudo usermod -aG grupo1 user2  
  
(aideelorenzo@aidee)-[~]  
$ sudo usermod -aG grupo3 user3  
  
(aideelorenzo@aidee)-[~]  
$ sudo usermod -aG grupo2 user1  
  
(aideelorenzo@aidee)-[~]  
$ sudo usermod -aG grupo2 user2
```

- **usermod:** Modifica un usuario existente.
- **-aG grupo1:** Agregar (a) dicho usuario a un grupo (G)

## Verificar los grupos de un usuario

Si queremos comprobar a qué grupo pertenece un usuario, el comando `id <nombre_usuario>` ayuda a dar con la información.

```
(aideelorenzo@aidee)-[~]
$ id user3
uid=1003(user3) gid=1003(user3) groups=1003(user3)

(aideelorenzo@aidee)-[~]
$ sudo usermod -aG grupo4 user2

(aideelorenzo@aidee)-[~]
$ sudo usermod -aG grupo5 user2

(aideelorenzo@aidee)-[~]
$ sudo usermod -aG grupo4 user1

(aideelorenzo@aidee)-[~]
$ id user2
uid=1005(user2) gid=1005(user2) groups=1005(user2),1006(grupo1),1007(grupo2),1009
(grupo4),1010(grupo5)

(aideelorenzo@aidee)-[~]
$ id user1
uid=1004(user1) gid=1004(user1) groups=1004(user1),1007(grupo2),1009(grupo4)

(aideelorenzo@aidee)-[~]
$
```

Para listar los grupos del sistema: `cat /etc/group`

```
(aideelorenzo@aidee)-[~]
$ tail /etc/group
aideelorenzo:x:1000:
kaboxer:x:136:aideelorenzo
user3:x:1003:
user1:x:1004:
user2:x:1005:
grupo1:x:1006:user2
grupo2:x:1007:user1,user2
grupo3:x:1008:user3
grupo4:x:1009:user2,user1
grupo5:x:1010:user2
```

**NOTA:** Se utilizó el comando `tail` para verificar los últimos grupos que han sido agregados. De utilizar `cat` (que también es válido), se vería solo el inicio de `/etc/group`. Se vería de la siguiente forma.

```

(aideelorenzo@aidee)-[~]
$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:aideelorenzo
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:

```

## Eliminación de un grupo

Si ya no se requiere tener un grupo, con **groupdel <nombre\_grupo>** se ejecuta su eliminación.

```

(aideelorenzo@aidee)-[~]
$ sudo groupdel grupo4

(aideelorenzo@aidee)-[~]
$ sudo groupdel grupo2

(aideelorenzo@aidee)-[~]
$ tail /etc/group
nm-openconnect:x:134:
kali-trusted:x:135:
aideelorenzo:x:1000:
kaboxer:x:136:aideelorenzo
user3:x:1003:
user1:x:1004:
user2:x:1005:
grupo1:x:1006:user2
grupo3:x:1008:user3
grupo5:x:1010:user2

(aideelorenzo@aidee)-[~]
$ id user2
uid=1005(user2) gid=1005(user2) groups=1005(user2),1006(grupo1),1010(grupo5)

(aideelorenzo@aidee)-[~]
$

```

## Eliminar un usuario de un grupo

Para evitar posibles errores, se recomienda eliminar primero a los usuarios del grupo antes de eliminar el grupo. Para eso existen dos formas:

### 1. Con `gpasswd -d <nombre_usuario> <nombre_grupo>`

```
(aideelorenzo@aidee)-[~]
$ id user2
uid=1005(user2) gid=1005(user2) groups=1005(user2),1006(grupo1),1010(grupo5)

(aideelorenzo@aidee)-[~]
$ sudo gpasswd -d user2 grupo1
Removing user user2 from group grupo1

(aideelorenzo@aidee)-[~]
$ id user2
uid=1005(user2) gid=1005(user2) groups=1005(user2),1010(grupo5)

(aideelorenzo@aidee)-[~]
$
```

### 2. Manualmente, editando el archivo `/etc/group`.

Para este segundo método utilizamos el comando **nano** para editar el archivo.

```
(aideelorenzo@aidee)-[~]
$ sudo nano /etc/group
```

Una vez dentro del archivo, buscamos el grupo correspondiente.

```
user1:x:1004:
user2:x:1005:
grupo1:x:1006:
grupo3:x:1008:user3
grupo5:x:1010:user2
```

Borramos el nombre de usuario hasta “:”

```
user2:x:1005:
grupo1:x:1006:
grupo3:x:1008:user3
grupo5:x:1010:

File Name to Write: /etc/group
^G Help M-D DOS Format
^C Cancel M-M Mac Format
```

Presionamos **Ctrl + X** para guardar los cambios, posteriormente **Y**, y damos enter para guardar los cambios y salir del archivo.

Al comprobar con **id** el usuario ya no debería tener un grupo asignado.

```
(aideelorenzo@aidee)-[~]
$ id user2
uid=1005(user2) gid=1005(user2) groups=1005(user2)
```

## Permisos de Archivos en Linux

### ¿Qué son los permisos en Linux?

Los permisos en Linux determinan quién puede leer, escribir o ejecutar un archivo o directorio. Se utilizan para proteger la información y restringir el acceso a usuarios no autorizados.

### Tipos de permisos

Cada archivo y directorio tiene tres tipos de permisos:

- **Lectura (r - Read):** Permite ver el contenido del archivo o listar el contenido de un directorio.
- **Escritura (w - Write):** Permite modificar el contenido de un archivo o agregar/eliminar archivos dentro de un directorio.
- **Ejecución (x - Execute):** Permite ejecutar un archivo si es un programa o script. En directorios, permite acceder a ellos y cambiar de ubicación.



## Asignación de permisos y valores numéricos

Cada permiso tiene un valor numérico asignado:

- **Lectura (r)** = 4
- **Escritura (w)** = 2
- **Ejecución (x)** = 1

Cada archivo en Linux tiene tres tipos de permisos asignados a tres tipos de usuarios:

- **Dueño (Owner - u)**: Usuario que creó el archivo.
- **Grupo (Group - g)**: Usuarios del mismo grupo que el dueño.
- **Otros (Others - o)**: Cualquier otro usuario en el sistema.

## Visualización de permisos en archivos

El comando `ls -l` mostrará los permisos de todos los archivos del directorio en el que estemos. Los permisos se presentarán en la primera columna.

```
(aideelorenzo@aidee)-[~/Permisos_Linux]
$ ls -l
total 0
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:29 archivo.txt
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:29 archivo2.txt
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:32 archivo8.txt
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:32 archivo_nuevo.txt
```

Para entender el desglose de los permisos, se facilita la siguiente tabla:

-	r	w	-	r	-	-	r	-	-
-	r	w	-	r	-	-	r	-	-
-	r	w	-	r	-	-	r	-	-

Primera columna: Indica que es un archivo (d para directorios)

Segunda columna: Permisos del dueño (lectura, escritura)

Tercera columna: Permisos del grupo (solo lectura)

Cuarta columna: Permisos para otros (solo lectura)

Para ver los permisos de un único archivo es necesario el comando `ls -l + <nombre_archivo>`

```
(aideelorenzo@aidee)-[~/Permisos_Linux]
$ ls -l archivo.txt
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:29 archivo.txt
```

Para mostrar los permisos de varios archivos, `ls -l + <nombre_archivo1> <nombre_archivo2>`

```
(aideelorenzo@aidee)-[~/Permisos_Linux]
$ ls -l archivo.txt archivo8.txt archivo_nuevo.txt
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:29 archivo.txt
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:32 archivo8.txt
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:32 archivo_nuevo.txt
```

- **rw- (Dueño):** Señala lectura y escritura para el creador del archivo.
- **r-- (Grupo):** Señala solo lectura para los usuarios del mismo grupo que el dueño del archivo.
- **r-- (Otros):** Señala solo lectura para los demás usuarios.

## Asignación de permisos en forma numérica

Para asignar permisos utilizamos el comando `chmod` + `<valor_numérico_permiso>` + `<nombre_archivo>`

```
(aideelorenzo@aidee)-[~/Permisos_Linux]
$ chmod 734 archivo.txt
File System
(aideelorenzo@aidee)-[~/Permisos_Linux]
$ ls -l archivo.txt archivo8.txt
-rwx-wxr-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:29 archivo.txt
-rw-r--r-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:32 archivo8.txt
```

**Explico:** Al asignar permisos de forma numérica, se puede aplicar un valor específico para cada permiso. Si se desea aplicar más de uno, se suman los valores correspondientes y, según la columna donde se coloque el número, se asigna el permiso al dueño, al grupo o a otros usuarios.

El valor 7 representa la suma de todos los permisos: lectura, escritura y ejecución. El valor 3 combina escritura y ejecución, mientras que 4 significa solo lectura.

**NOTA:** Si un archivo cambia de color en la terminal, como a verde, es porque se le ha asignado el permiso de ejecución (x), lo que lo convierte en un archivo ejecutable.

```
(aideelorenzo@aidee)-[~/Permisos_Linux]
$ chmod 070 archivo8.txt
(aideelorenzo@aidee)-[~/Permisos_Linux]
$ ls -l archivo8.txt
---rwx--- 1 aideelorenzo aideelorenzo 0 Feb  6 18:32 archivo8.txt
(aideelorenzo@aidee)-[~/Permisos_Linux]
$
```

Con el mismo comando se pueden asignar permisos a directorios. En el ejemplo se aplican permisos totales a quien creó el directorio **Permisos\_Linux**.

```
(aideelorenzo@aidee)-[~]
$ ls -l
total 36
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Desktop
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Documents
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Downloads
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Music
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 18:32 Permisos_Linux
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  5 20:43 Pictures
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Public
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Templates
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Videos

(aideelorenzo@aidee)-[~]
$ chmod 700 Permisos_Linux

(aideelorenzo@aidee)-[~]
$ ls -l
total 36
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Desktop
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Documents
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Downloads
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Music
drwx----- 2 aideelorenzo aideelorenzo 4096 Feb  6 18:32 Permisos_Linux
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  5 20:43 Pictures
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Public
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Templates
drwxr-xr-x 2 aideelorenzo aideelorenzo 4096 Feb  6 02:35 Videos
```

## Asignación de permisos en forma alfabética

También se pueden asignar permisos de la siguiente forma, indicando a qué columna prefieres asignar los permisos.

```
(aideelorenzo@aidee)-[~/Permisos_Linux]
$ ls -l archivo8.txt
-rwx-- 1 aideelorenzo aideelorenzo 0 Feb  6 18:32 archivo8.txt

(aideelorenzo@aidee)-[~/Permisos_Linux]
$ chmod u=wr,g=rx,o=w archivo8.txt

(aideelorenzo@aidee)-[~/Permisos_Linux]
$ ls -l archivo8.txt
-rw-r-x-w- 1 aideelorenzo aideelorenzo 0 Feb  6 18:32 archivo8.txt

(aideelorenzo@aidee)-[~/Permisos_Linux]
$
```

## Conclusión y reflexión final

### Resumen de lo aprendido y reflexión personal

Antes de iniciar este proyecto tomé en cuenta la investigación en diversas fuentes sobre los temas expuestos. Linux es un sistema operativo usado por pocos usuarios comunes que, a la hora de pasarse a un nuevo entorno no comprenden la administración de ciertos recursos. El proyecto plantea el proceso que llevó a cabo desde la creación de archivos hasta la administración de permisos de los mismos. Con la utilización de comando como: **useradd**, **groupadd**, **userdel**, **tail**, **cat**, **chmod** y otros más, la gestión sobre este tipo de recursos se hace más sencilla. Con la combinación de diversas técnicas lo aprendido se realiza en poco tiempo, es cuestión de saber analizar las distintas situaciones que puede presentarse según las necesidades de una empresa o un usuario.

En un punto de vista personal y profesional, la gestión de usuarios y permisos es tan importante como cualquier proceso de administración de un sistema. Es la barrera de seguridad que hay entre el usuario y la información confidencial o de libre acceso. No se debe pasar desapercibido la asignación de permisos necesarios a los usuarios correspondientes, de no hacerlo las consecuencias pueden no ser las mejores.

Este proyecto me ha ayudado a ver a profundidad lo indispensable que es manejar este tipo de acciones en un entorno Linux. Manejar permisos desde la terminal ayuda con la flexibilidad y manejo de herramientas nativas del sistema. Aprender lo básico no solo es necesario para pasar a lo demás, es importante saber que, más adelante será la base para dominar técnicas más especializadas.

## Importancia del tema

La gestión de permisos de archivos y la administración de usuarios y grupos en Linux son fundamentales para garantizar la seguridad y eficiencia de cualquier sistema. Estos aspectos permiten controlar el acceso a los recursos, asegurando que solo los usuarios autorizados puedan modificar, leer o ejecutar archivos específicos. Una correcta configuración no solo protege el sistema de posibles accesos no deseados, sino que también optimiza la gestión de recursos y facilita la resolución de problemas de manera más eficiente. En entornos corporativos o servidores, donde múltiples usuarios requieren diferentes niveles de acceso, comprender cómo gestionar estos permisos es esencial para mantener la integridad y el buen funcionamiento del sistema. Esta habilidad es clave para los administradores de sistemas que buscan garantizar la confidencialidad, integridad y disponibilidad de la información.