1. Introducción

a. ¿Qué es un usuario?

Un **usuario** en Linux es una identidad asociada a una cuenta en el sistema operativo. Cada usuario tiene un nombre único que permite diferenciarlo y acceder a recursos específicos del sistema. Los usuarios pueden ser:

- **Humanos**, como los administradores o usuarios regulares.
- **Servicios**, representados por cuentas de usuario especiales asignadas a procesos o aplicaciones.

b. Diferencias entre usuario y root

- **Usuario estándar**:
  - Permisos limitados: Puede ejecutar tareas normales, como editar sus propios archivos.
  - No puede realizar cambios en configuraciones del sistema sin autorización.
- **Usuario root**:
  - Es el superusuario con privilegios completos.
  - Puede ejecutar cualquier comando y realizar modificaciones críticas en el sistema.
  - **Cuidado**: Usar `root` de manera irresponsable puede comprometer el sistema.

c. Importancia del usuario en Linux

El diseño multiusuario de Linux es clave para:

- Proteger la privacidad y seguridad de los datos.
- Distribuir tareas y accesos según roles.
- Mantener un entorno organizado y eficiente.

2. Comandos básicos

a. `whoami`

Este comando muestra el nombre del usuario con el que se ha iniciado sesión:

```
whoami
```

b. `id`

Proporciona información detallada del usuario, como su UID, GID y grupos asociados:

```
id
```

c. `adduser`

Se utiliza para añadir un usuario al sistema. Este comando es interactivo, más fácil de usar que `useradd`:

```
sudo adduser nombre_usuario
```

Solicita información como la contraseña, nombre completo y directorio principal.

d. `usermod`

Permite modificar los atributos de un usuario existente. Ejemplo:

- Cambiar el shell del usuario:

  ```
  sudo usermod -s /bin/zsh nombre_usuario
  ```

- Mover el directorio personal del usuario:

  ```
  sudo usermod -d /nuevo_directorio -m nombre_usuario
  ```

3. Grupos

a. ¿Qué es un grupo?

Un **grupo** en Linux es una colección de usuarios que comparten permisos comunes para facilitar la gestión de accesos a archivos y recursos.

b. `groups`

Muestra los grupos a los que pertenece un usuario:

```
groups nombre_usuario
```

c. `groupadd`

Se utiliza para crear un nuevo grupo:

```
sudo groupadd nombre_grupo
```

d. `groupdel`

Elimina un grupo del sistema, siempre que no tenga usuarios asociados:

```
sudo groupdel nombre_grupo
```

e. `usermod -aG`

Añade un usuario a un grupo sin eliminarlo de otros grupos existentes:

```
sudo usermod -aG nombre_grupo nombre_usuario
```

f. `id`

El comando `id` también muestra los grupos a los que pertenece un usuario:

```
id nombre_usuario
```

4. Permisos y privilegios

a. `rwx`

Los permisos en Linux se dividen en:

- **r (read)**: Lectura.
- **w (write)**: Escritura.
- **x (execute)**: Ejecución. Se aplican para **usuario (u)**, **grupo (g)** y **otros (o)**. Ejemplo:

```
-rwxr--r--
```

Significa que el usuario puede leer, escribir y ejecutar, pero el grupo solo puede leer y otros también solo pueden leer.

b. `chmod`

Cambia los permisos de archivos y directorios. Ejemplo:

- Usando formato simbólico:

  ```
  chmod u+x archivo
  ```

- Usando formato numérico:

  ```
  chmod 755 archivo
  ```

c. `chown`

Cambia el propietario de un archivo o directorio:

```
sudo chown nuevo_usuario archivo
```

d. `chgrp`

Cambia el grupo al que pertenece un archivo o directorio:

```
sudo chgrp nuevo_grupo archivo
```

e. Uso del `sudo`

El comando `sudo` permite a los usuarios ejecutar comandos con privilegios administrativos. Ejemplo:

```
sudo apt update
```

Solo los usuarios autorizados (definidos en el archivo `/etc/sudoers`) pueden utilizar este comando.

5. Configuración avanzada

a. `chage`

Configura políticas de contraseñas avanzadas, como la expiración:

- Configurar una fecha de expiración para la cuenta:

  ```
  sudo chage -E YYYY-MM-DD nombre_usuario
  ```

- Ver el estado de la cuenta:

  ```
  sudo chage -l nombre_usuario
  ```