

# Permisos de archivos en Linux

## Descripción del Proyecto

Este proyecto tiene como objetivo **establecer y gestionar los permisos de acceso** entre los distintos roles de un equipo de trabajo, con el fin de **mantener la confidencialidad, integridad y seguridad de los datos** contenidos en diversos archivos y directorios.

## Verificación de Permisos en Archivos y Directorios

Se utilizó el comando `ls -l` para listar los permisos actuales de los archivos. A continuación, se detalla el estado de los permisos:

```
researcher2@2255d014d046:~/projects$ ls -l
total 20
drwx--x--- 2 researcher2 research_team 4096 May 14 16:30 drafts
-rw-rw-rw- 1 researcher2 research_team  46 May 14 16:30 project_k.txt
-rw-r----- 1 researcher2 research_team  46 May 14 16:30 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 May 14 16:30 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 May 14 16:30 project_t.txt
researcher2@2255d014d046:~/projects$
```

## Permisos Iniciales

Archivo	Usuario (User)	Grupo (Group)	Otros (Other)
project_k.txt	read, write	read, write	read, write
project_m.txt	read, write	read	ninguno
project_r.txt	read, write	read, write	read
project_t.txt	read, write	read, write	read

.project\_x. read, write write ninguno  
txt

## Explicación de la Cadena de Permisos

- **r** → Permiso de lectura (read): permite visualizar el contenido del archivo.
- **w** → Permiso de escritura (write): permite modificar el contenido del archivo.
- **x** → Permiso de ejecución (execute): permite ejecutar el archivo (en el caso de scripts o binarios).
- **-** → Indica ausencia de ese permiso.

## Cambios Realizados en los Permisos

### 1. Eliminar permisos a otros usuarios

Para proteger archivos sensibles y limitar el acceso solo a roles autorizados, se ejecutaron los siguientes comandos:

```
chmod o-rw project_k.txt  
chmod o-r project_r.txt
```

```
researcher2@2255d014d046:~/projects$ chmod o-rw project_k.txt  
researcher2@2255d014d046:~/projects$  
researcher2@2255d014d046:~/projects$ chmod o-r project_r.txt  
researcher2@2255d014d046:~/projects$  
researcher2@2255d014d046:~/projects$ ls -l  
total 20  
drwx--x--- 2 researcher2 research_team 4096 May 14 16:30 drafts  
-rw-rw---- 1 researcher2 research_team  46 May 14 16:30 project_k.txt  
-rw-r----- 1 researcher2 research_team  46 May 14 16:30 project_m.txt  
-rw-rw---- 1 researcher2 research_team  46 May 14 16:30 project_r.txt  
-rw-rw-r-- 1 researcher2 research_team  46 May 14 16:30 project_t.txt  
researcher2@2255d014d046:~/projects$
```

- En **project\_k.txt**, se eliminaron los permisos de lectura y escritura para los "otros" usuarios.

- En `project_r.txt`, se revocó únicamente el permiso de lectura para "otros".

## 2. Gestión de archivos ocultos

Se utilizó el comando `ls -la` para listar también los archivos ocultos. Se identificó que el archivo `.project_x.txt` tenía permisos de escritura para el usuario y el grupo, lo cual no era deseado debido a su naturaleza sensible.

```
researcher2@2255d014d046:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 May 14 16:30 .
drwxr-xr-x 3 researcher2 research_team 4096 May 14 16:53 ..
-rw--w---- 1 researcher2 research_team   46 May 14 16:30 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 May 14 16:30 drafts
-rw-rw---- 1 researcher2 research_team   46 May 14 16:30 project_k.txt
-rw-r----- 1 researcher2 research_team   46 May 14 16:30 project_m.txt
-rw-rw---- 1 researcher2 research_team   46 May 14 16:30 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 May 14 16:30 project_t.txt
```

Para corregirlo, se ejecutó:

```
chmod u-w,g-w .project_x.txt
```

```
researcher2@2255d014d046:~/projects$ chmod u-w,g-w .project_x.txt
researcher2@2255d014d046:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 May 14 16:30 .
drwxr-xr-x 3 researcher2 research_team 4096 May 14 16:53 ..
-r----- 1 researcher2 research_team   46 May 14 16:30 .project_x.txt
drwx--x--- 2 researcher2 research_team 4096 May 14 16:30 drafts
-rw-rw---- 1 researcher2 research_team   46 May 14 16:30 project_k.txt
-rw-r----- 1 researcher2 research_team   46 May 14 16:30 project_m.txt
-rw-rw---- 1 researcher2 research_team   46 May 14 16:30 project_r.txt
-rw-rw-r-- 1 researcher2 research_team   46 May 14 16:30 project_t.txt
```

Con esto, se eliminó el permiso de escritura tanto al propietario como al grupo.

## 3. Cambio de permisos en un directorio

Para restringir la ejecución de un directorio llamado `drafts`, se ejecutó:

```
chmod u-x,g-x drafts
```

```
researcher2@2255d014d046:~/projects$ chmod u-x,g-x drafts
researcher2@2255d014d046:~/projects$ ls -la
total 32
drwxr-xr-x 3 researcher2 research_team 4096 May 14 16:30 .
drwxr-xr-x 3 researcher2 research_team 4096 May 14 16:53 ..
-r----- 1 researcher2 research_team  46 May 14 16:30 .project_x.txt
-rw----- 2 researcher2 research_team 4096 May 14 16:30 drafts
-rw-rw---- 1 researcher2 research_team  46 May 14 16:30 project_k.txt
-rw-r----- 1 researcher2 research_team  46 May 14 16:30 project_m.txt
-rw-rw---- 1 researcher2 research_team  46 May 14 16:30 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 May 14 16:30 project_t.txt
researcher2@2255d014d046:~/projects$
```

Esto impide que tanto el usuario como los miembros del grupo puedan acceder al contenido del directorio mediante navegación (`cd`), reforzando así la seguridad en entornos de trabajo colaborativo.

## Resumen

En este ejercicio se aplicaron principios básicos de **seguridad de archivos en sistemas Unix/Linux**, centrados en la gestión de permisos mediante el comando `chmod`. Se fortaleció la privacidad y el control de acceso en función de roles, lo cual es esencial en entornos profesionales donde el trabajo colaborativo involucra datos sensibles.