ACTIVIDADES FINALES

Taula de continguts

Ejercicio 1. Gestión de usuarios en Linux.

a) Crea los grupos SI1 y SI2

```
anna@SIAnna:~$ sudo groupadd SI1
[sudo] contraseña para anna:
anna@SIAnna:~$ sudo groupadd SI2
anna@SIAnna:~$
```

B) Crea el usuario0 cuyo home sea "/home/usuario_cero", copiando los archivos de configuración de "/etc/skel". Comprueba a cuántos grupos pertenece.

```
anna@SIAnna:~$ sudo useradd -m -d /home/usuario_cero -s /bin/bash usuario0
anna@SIAnna:~$ groups usuario0
```

c) Localiza la línea de dicho usuario en el fichero "/etc/passwd" para comprobar sus datos

Usamos cat para mostrar el fichero y grep para filtrar

```
anna@SIAnna:~$ cat /etc/passwd | grep usuario0
usuario0:x:1002:1004::/home/usuario_cero:/bin/bash
anna@SIAnna:~$
```

d) Elimina el usuario0 y su carpeta home
para borrar el usuario ponemos userdel para borrar el usuario y -r que sirve para borrar su directorio

```
anna@SIAnna:~$ sudo userdel -r usuario0
[sudo] contraseña para anna:
```

e) Crea los usuarios usuario1 y usuario2. Ambos deben pertenecer únicamente al grupo SI1 como principal. Comprueba su pertenencia a dicho grupo.
Creamos los usuarios con useradd y ponemos -g para poder especificarle el grupo

```
anna@SIAnna:~$ sudo useradd -m -g SI1 usuario1
anna@SIAnna:~$ sudo useradd -m -g SI1 usuario2
anna@SIAnna:~$ groups usuario1
usuario1 : SI1
anna@SIAnna:~$ groups usuario2
usuario2 : SI1
anna@SIAnna:~$
```

f) Crea los usuarios usuario3 y usuario4, perteneciendo como grupo principal a SI2

```
anna@SIAnna:~$ sudo useradd -m -g SI2 usuario3
anna@SIAnna:~$ sudo useradd -m -g SI2 usuario4
```

g) Modifica el Shell por defecto del usuario3 a "/bin/sh"
por defecto ya esta *en bin* /sh

```
anna@SIAnna:~$ sudo usermod -s /bin/sh usuario3
[sudo] contraseña para anna:
usermod: sin cambios
```

h) Intenta eliminar SI1.
No se puede eliminar porque tiene usuarios designados

```
anna@SIAnna:~$ sudo groupdel SI1
groupdel: no se pudo eliminar el grupo primario del usuario «usuario1»
anna@SIAnna:~$
```

Ejercicio 2. Seguridad en cuentas de usuario en Linux:

a) Suministra las contraseña sistEmas_%20 a todos los usuarios antes creados.

```
anna@SIAnna:~$ sudo passwd usuario1
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
anna@SIAnna:~$ sudo passwd usuario2
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
anna@SIAnna:~$ sudo passwd usuario3
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
anna@SIAnna:~$ sudo passwd usuario4
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
anna@SIAnna:~$
```

b) Deshabilita la cuenta de usuario3 y bloquea la contraseña de usuario4. Compruébalo.

```
anna@SIAnna:~$ sudo usermod -L usuario4
```

```
anna@SIAnna:~$ sudo usermod -s /sbin/nologin usuario3
```

Ahora lo comprobamos:

```
anna@SIAnna:~$ sudo passwd -S usuario3
usuario3 P 2025-02-19 0 99999 7 -1
anna@SIAnna:~$ sudo passwd -S usuario4
usuario4 L 2025-02-19 0 99999 7 -1
anna@SIAnna:~$
```

c) Habilita la cuenta de usuario3 y desbloquea la contraseña de usuario4. Compruébalo.
-U → bloquea la cuenta
-s /bin /bash vuelve a permitir el acceso

```
anna@SIAnna:~$ sudo usermod -s /bin/bash usuario3
anna@SIAnna:~$ sudo usermod -U usuario4
```

Ahora comprobamos

```
anna@SIAnna:~$ sudo passwd -S usuario4
usuario4 P 2025-02-19 0 99999 7 -1
anna@SIAnna:~$ sudo passwd -S usuario3
usuario3 P 2025-02-19 0 99999 7 -1
anna@SIAnna:~$
```

d) Establece 20 días como máximo y 10 días como mínimo para cambiar la contraseña de usuario2. Compruébalo.

```
anna@SIAnna:~$ sudo chage -l usuario2
Último cambio de contraseña                                   : feb 19, 2025
La contraseña caduca                                  : mar 11, 2025
Contraseña inactiva                                   : nunca
La cuenta caduca                                              : nunca
Número de días mínimo entre cambio de contraseña              : 10
Número de días máximo entre cambio de contraseña              : 20
Número de días de aviso antes de que caduque la contraseña    : 7
anna@SIAnna:~$
```

e) Establece una fecha para usuario4, a partir de la cual la cuenta caducará y será inaccesible. Compruébalo. Elimina la expiración de la cuenta y compruébalo

```
anna@SIAnna:~$ sudo chage -E 2025-03-01 usuario4
anna@SIAnna:~$ sudo chage -l usuario4
Último cambio de contraseña                                   : feb 19, 2025
La contraseña caduca                                          : nunca
Contraseña inactiva                                           : nunca
La cuenta caduca                                              : mar 01, 2025
Número de días mínimo entre cambio de contraseña              : 0
Número de días máximo entre cambio de contraseña              : 99999
Número de días de aviso antes de que caduque la contraseña    : 7
anna@SIAnna:~$ sudo chage -E -1 usuario4
anna@SIAnna:~$
```

Ejercicio 3. Gestión de recursos y permisos en Linux y Windows

. En Linux:

a) En tu home, crea la carpeta dirPerm y, dentro de ella, un archivo llamado permisos.

```
anna@SIAnna:~$ mkdir ~/dirPerm
anna@SIAnna:~$ touch ~/dirPerm/permisos
```

b) Emplea la notación octal para modificar los permisos de dirPerm a rwxr- - - - -. Y la notación simbólica para deshabilitar para el grupo el permiso de lectura sobre el archivo permisos. Compruébalo.

chmod 740 ~/dirPerm →
7 (rwx) → Propietario tiene todos los permisos.
4 (r--) → Grupo solo tiene lectura.
0 (---) → Otros no tienen permisos.
chmod g-r ~/dirPerm/permisos → Quita el permiso de lectura al grupo.

```
anna@SIAnna:~$ chmod 740 ~/dirPerm
anna@SIAnna:~$ chmod g-r ~/dirPerm/permisos
anna@SIAnna:~$ ls -l ~/dirPerm/
total 0
-rw--w-r-- 1 anna anna 0 feb 19 19:54 permisos
anna@SIAnna:~$
```

c) Emplea la notación octal para modificar los permisos de dirPerm a rwxrwxrw-. Y la notación simbólica para habilitar todos los permisos para el propietario y el grupo, y deshabilitar todos los permisos al resto de usuarios sobre el archivo permisos. Compruébalo.

chmod 776 ~/dirPerm →
7 (rwx) → Propietario y grupo tienen todos los permisos.
6 (rw-) → Otros pueden leer y escribir.
chmod u+rwx,g+rwx,o-rwx ~/dirPerm/permisos →

Propietario y grupo tienen permisos totales.
Otros no tienen acceso.

```
anna@SIAnna:~$ chmod 776 ~/dirPerm
anna@SIAnna:~$ chmod u+rwx,g+rwx,o-rwx ~/dirPerm/permisos
anna@SIAnna:~$ ls -l ~/dirPerm
total 0
-rwxrwx--- 1 anna anna 0 feb 19 19:54 permisos
anna@SIAnna:~$
```

d) Cambia el propietario y grupo de dirPerm a usuario1 y SI1, respectivamente, afectando a su contenido.

chown -R usuario1 ~/dirPerm → Cambia el propietario a usuario1.
chgrp -R SI1 ~/dirPerm → Cambia el grupo a SI1.
-R → Aplica los cambios a todos los archivos dentro de dirPerm.

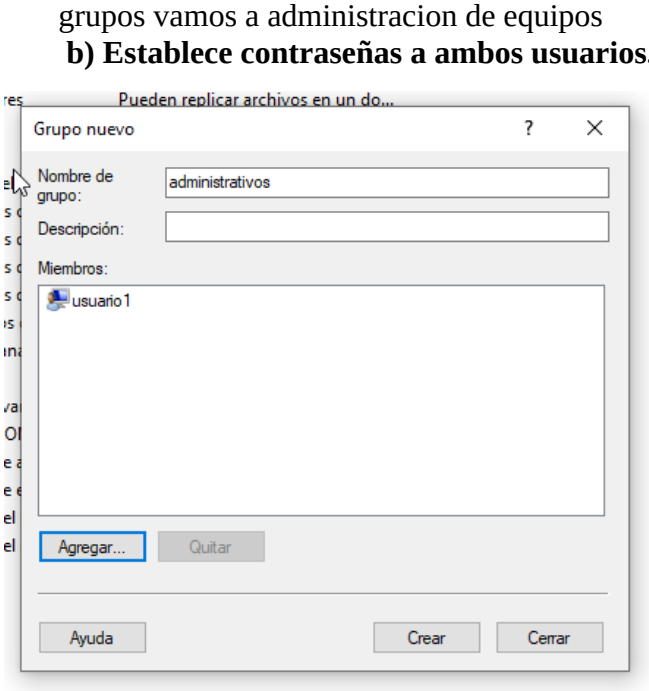
```
anna@SIAnna:~$ sudo chown -R usuario1 ~/dirPerm
[sudo] contraseña para anna:
anna@SIAnna:~$ sudo chgrp -R SI1 ~/dirPerm
anna@SIAnna:~$
```

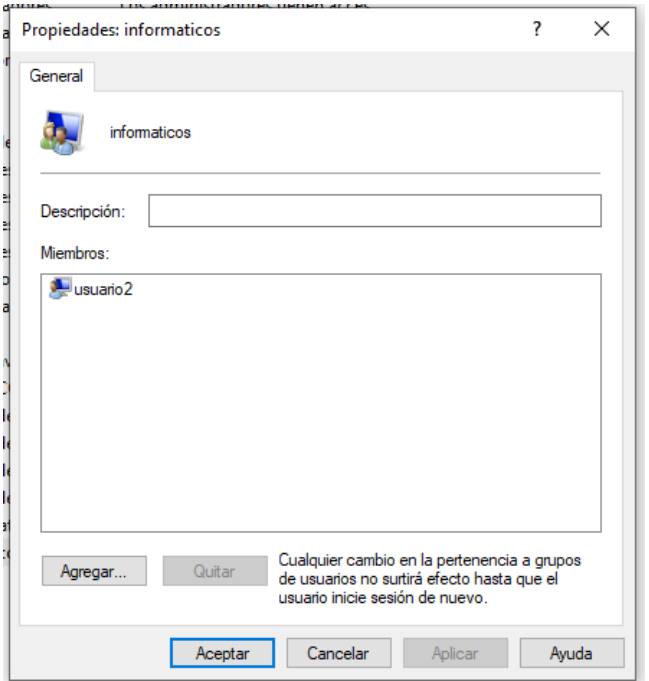
En Windows:

a) Crea dos usuarios, usuario1 y usuario2, y dos grupos de usuarios, Administrativos e Informáticos. Asigna cada usuario a un grupo. Compruébalo.

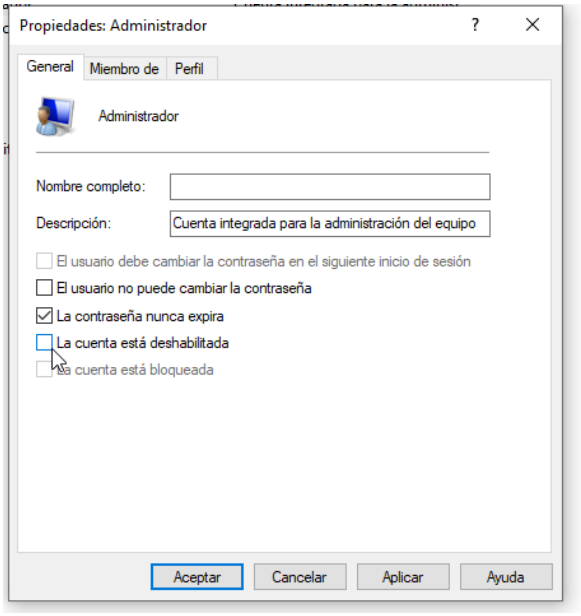
Vamos a configuracion y a familia y otros grupos, ahi podremos crear los usuarios y para crear los grupos vamos a administracion de equipos

b) Establece contraseñas a ambos usuarios.



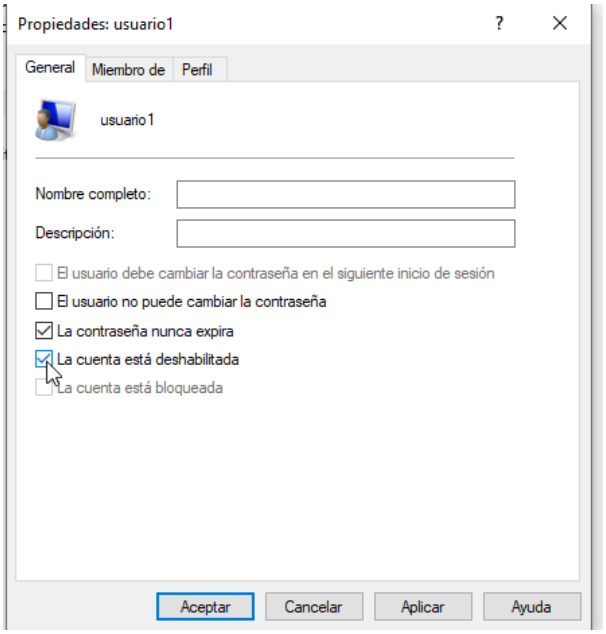


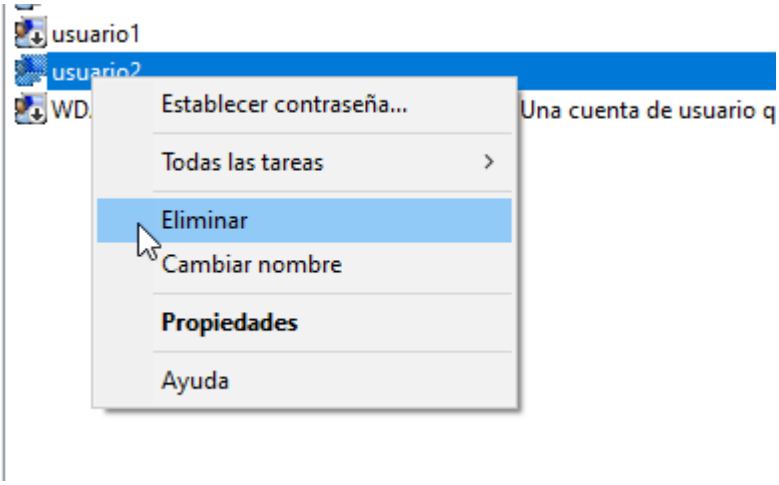
c) Habilita el usuario Administrador.
Vamos al usuario administrador en el menu de usuarios y quitamos que la cuenta esta dehabilitada importante darle a aplicar



d) Deshabilita el usuario usuario1. Elimina usuario2.

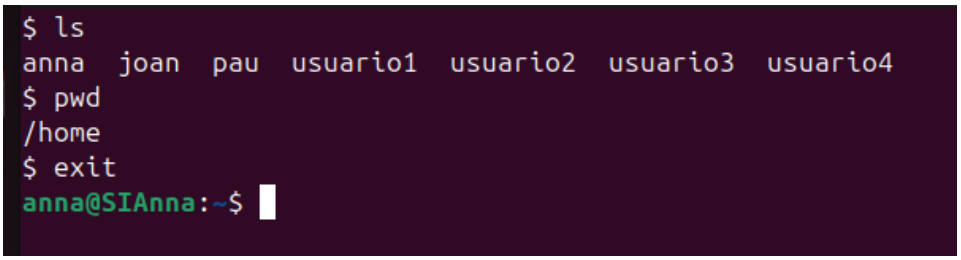
Aqui desabilitamos el usuario y para eliminarlo le damos clic derecho a eliminarlo



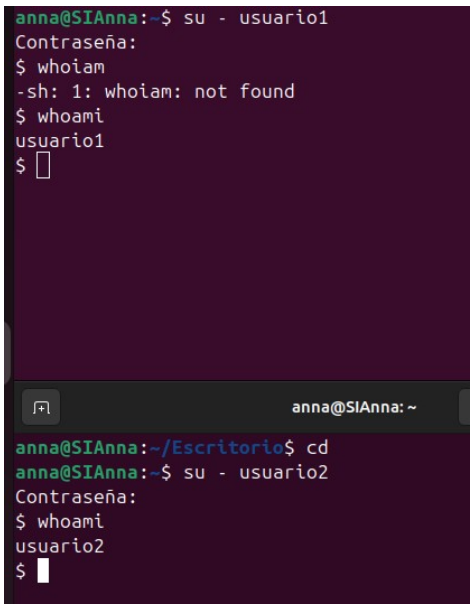


Ejercicio 4. Acceso de usuarios en Linux.

a) Desde el terminal, ejecuta una sesión como usuario1. Lista el contenido de su home. Muestra el valor de la variable PWD. Sal de dicha sesión.



b) Accede desde varios terminales con diferentes usuarios. Desde cualquiera de ellos, muestra los usuarios que se encuentran conectados.



Ejercicio 5. Permisos especiales en Linux.

a) Crea un directorio llamado dirCompartido en el que ahora modifica los permisos del directorio para que solo el propietario de cada objeto pueda borrar sus propios archivos.

mkdir /dirCompartido → Crea la carpeta en la raíz del sistema.
chown :SI2 /dirCompartido → Asigna el grupo SI2 como propietario del directorio.
chmod 3770 /dirCompartido → Configura los permisos:

3 (setgid y sticky bit):
setgid (2) → Archivos creados dentro heredarán el grupo SI2.

sticky bit (1) → Solo el propietario de un archivo puede borrarlo, aunque otros tengan permisos de escritura.
770 (rwxrwx---) → Propietario y grupo tienen permisos totales; otros no tienen acceso.

```
anna@SIAnna:~$ sudo chown :SI2 dirCompartido
anna@SIAnna:~$ sudo chmod 3770 dirCompartido
```

b) Ahora modifica los permisos de dirCompartido para que los archivos y subdirectorios creados en su interior sean forzados a pertenecer al grupo del directorio y no al grupo del usuario que lo haya creado.

```
anna@SIAnna:~$ sudo chmod g+s dirCompartido
```

chmod g+s → Activa el bit setgid, lo que obliga a que todos los archivos creados en dirCompartido pertenezcan al grupo del directorio (SI2) en lugar del grupo del usuario creador.

Ejercicio 6. Máscara de permisos en Linux.

a) Para usuario1, modifica la máscara de permisos por defecto para que los ficheros creados tengan todos los permisos activos. Comprueba si es posible.

Sí es posible modificar temporalmente la umask, pero se restablecerá al cerrar sesión.

b) Configura la máscara de permisos por defecto para que en los nuevos ficheros el propietario tenga permiso de lectura solo, y los demás no tengan ningún permiso. Sal de la sesión de usuario1 y comprueba si se ha guardado la máscara de permisos. ¿Podríamos hacerla permanente?

```
$ umask 077
$ touch archivo_restringido
$ ls -l archivo_restringido
-rw------- 1 usuario1 SI1 0 feb 21 12:58 archivo_restringid
o
$
```

```
anna@SIAnna:~$ nano ~/.bashrc
```

Ejercicio 7. Superusuario en Linux.

¿Por qué la cuenta root viene deshabilitada por defecto?¿Cómo se podría habilitar?

por razones de seguridad. En lugar de iniciar sesión directamente como root, se usa sudo para ejecutar comandos con privilegios elevados.

Razones por las que root está deshabilitado:

Evita errores críticos: Un usuario con acceso total podría eliminar archivos esenciales del sistema accidentalmente.
Mayor seguridad: Si un atacante obtiene acceso al sistema, necesitará una contraseña adicional para usar sudo, lo que dificulta la escalada de privilegios.
Mejor control: sudo registra todos los comandos ejecutados con privilegios elevados en /var/log/auth.log, permitiendo auditoría.

Ejercicio 8. Alias y variables en Linux.

a) Muestra, a través de variables el directorio de trabajo actual, el nombre del equipo, el shell y el login del usuario actual.
$PWD → Muestra el directorio de trabajo actual.
$HOSTNAME → Indica el nombre del equipo.
$SHELL → Muestra el shell que usa el usuario.
$USER → Muestra el nombre del usuario actual.

b) Crea un alias que permita crear nuevos ficheros con la cadena cfch. Compruébalo y elimina este.

```
anna@SIAnna:~$ alias cfch='touch nuevo_fichero'
```

```
anna@SIAnna:~$ unalias cfch
```

c) Crea una variable global llamada SISTEMAS con valor SI_23. Compruébalo.

```
^Canna@SIAnna:~export SISTEMAS="SI_23"3"
anna@SIAnna:~$ echo $SISTEMAS
SI_23
```

d) Haz permanente la variable global SISTEMAS.

```
  GNU nano 7.2          /home/anna/.bashrc *
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_comple
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
export SISTEMAS="SI_23"
```

```
anna@SIAnna:~$ source ~/.bashrc
anna@SIAnna:~$ echo $SISTEMAS
SI_23
anna@SIAnna:~$
```