

**MP0483.**

**Sistemas informáticos**

**UF4. Configuración de  
sistemas operativos**

**4.3. Perfiles y recursos**

# Índice

---

☰	Objetivos	3
☰	El perfil de usuario	4
☰	Directorios y ficheros implicados	8
☰	Información del perfil de usuario en Windows	11
☰	Información de perfil de usuario en Linux	16
☰	Cambiar la ruta por defecto del usuario	19
☰	Acceso a recursos y permisos locales	22
☰	Permisos en Windows	24
☰	Permisos en Linux	28
☰	Cómo cambiar los permisos de un archivo en Linux	32
☰	Ejemplos de permisos especiales en Linux	35
☰	Resumen	39

# Objetivos

---

Una óptima gestión de los recursos del sistema implica tener clara su organización y cómo se va a permitir a los usuarios disponer de esos recursos, estableciendo permisos y políticas de acceso que limiten la capacidad de actuación cuando sea necesario. De esta forma se salvaguarda la seguridad del sistema y a la vez se asignan los recursos de forma lógica a aquellas cuentas de usuario que realmente los necesitan.

En esta unidad perseguimos los siguientes objetivos:

1

Aprender a configurar los perfiles locales de los usuarios para permitir su trabajo de forma óptima, maximizando la seguridad del sistema.

2

Establecer permisos y políticas de acceso que limiten la capacidad de actuación como forma de salvaguardar la seguridad del sistema.

3

Conocer cómo utilizar algunos ficheros de configuración para gestionar el acceso y utilización de las cuentas de usuario.

---

¡Ánimo y adelante!

# El perfil de usuario

---

Aunque alguna vez se habla de ambos indistintamente, **el perfil de usuario no es lo mismo que la cuenta de usuario.**

---

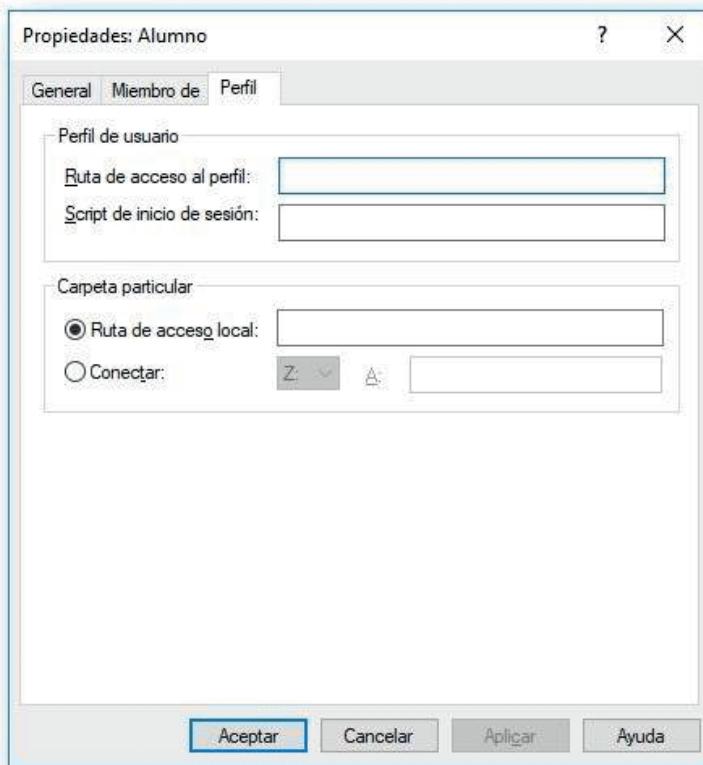
El perfil de usuario es el **conjunto de características** asociadas a una cuenta que configuran su entorno de trabajo en el sistema.

Este perfil de usuario incluye, por ejemplo, todos los parámetros de configuración específicos del entorno: aplicaciones autorizadas, conexiones de red, acceso a recursos compartidos, tipo de escritorio, cuotas de espacio asignado en disco, etc.

El perfil de usuario normalmente **se define al crear el usuario**, bien de forma predeterminada, modificada manualmente por el administrador, o por el hecho de asociarle a algún grupo de usuarios con determinados privilegios. En muchos casos, el perfil se termina de crear la primera vez que el usuario inicia sesión en el sistema.

El usuario podrá personalizar algunas características de su perfil, generalmente asociadas al aspecto de su escritorio y al modo en el que se muestra la información.

El sistema archivará en una carpeta la configuración del perfil del usuario. Por ejemplo, en el caso de Windows es en la carpeta "C:\Documents and Settings\All Users" (que pertenece al sistema y está oculta), de forma que en el siguiente inicio de sesión se cargue la configuración actualizada. De todas formas podremos acceder a la información de usuario a través del entorno gráfico, en la pantalla de la figura que veremos más adelante.



## Tipos de perfiles de usuario

En algunos sistemas operativos se pueden crear perfiles de varios tipos. Más allá de que puedas usarlos ahora o solamente trabajemos como usuarios locales, conviene tener claras las diferencias entre ellos.

### PERFIL DE USUARIO LOCAL

Se crea la primera vez que el usuario accede al sistema y **se almacena en el disco duro local**. Si se producen modificaciones sobre el perfil se harán localmente y no afectan al acceso del usuario en otros equipos.

### PERFIL DE USUARIO MÓVIL

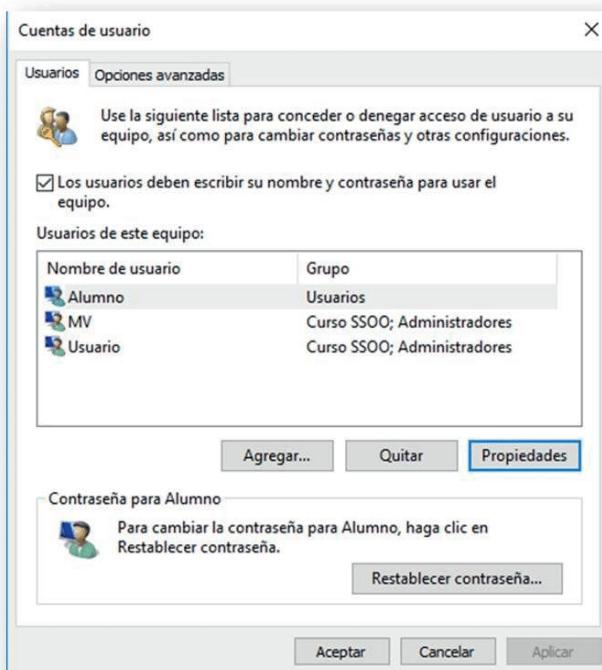
Está orientado al trabajo con servidores; el perfil es **creado por el administrador del sistema y se guarda en un servidor**. Cuando el usuario inicia sesión en el sistema su perfil se descarga desde el servidor y, si se producen modificaciones, se actualizan tanto localmente como en el servidor. Los perfiles de usuarios móviles estarán disponibles en el servidor cuando se inicie sesión como usuario de un dominio determinado. Si por alguna razón no están disponibles se puede crear un perfil local para usarlo en ese equipo.

## PERFIL OBLIGATORIO DE USUARIO

En este caso el perfil **permanece estable tal como ha sido definido por el administrador**. Se descarga al inicio de la sesión, pero no se actualiza al cerrarla, sino que simplemente se elimina del equipo local y permanece en el servidor. Esto permite que en cada inicio de sesión el entorno de trabajo aparezca “limpio”, tal como lo define el administrador, haciendo que el entorno de trabajo sea homogéneo para ese conjunto de usuarios.

## Ventajas del uso de perfiles de usuario

- Varios usuarios pueden utilizar el mismo equipo con entornos de trabajo diferentes.
- Cuando el usuario inicia una nueva sesión en un equipo se carga la configuración que tenía cuando finalizó la anterior (en el mismo u otro equipo).
- La personalización del escritorio que realiza un usuario es personal y no afecta a la de otros usuarios.
- Si se usan perfiles obligatorios el administrador del sistema se asegura de que siempre se inicien sesiones con la misma configuración, ya que los cambios son borrados al finalizar.



## ¿Qué es un “dominio”?

Un **dominio** está formado por un **conjunto de equipos conectados en red y administrados desde un punto centralizado (“controlador de dominio”)**. Cuando las redes son muy grandes pueden estar divididas en **“subdominios”** y existir varios controladores diferentes, que a su vez pueden depender de otro de mayor jerarquía.

Los controladores de dominio pueden realizar tareas como resolver direcciones (DNS), almacenar carpetas de información de los usuarios, acceso a aplicaciones comunes, automatización de copias de seguridad, etc.

---

**No debemos confundir este concepto con el dominio de Internet, asociado a la jerarquía de DNS para la traducción de nombres a direcciones IP.**

## Directarios y ficheros implicados

---

Los directorios y archivos relacionados con la información de los perfiles de usuario serán diferentes de un sistema operativo a otro, al igual que su ubicación.

En general podemos decir que estos ficheros almacenarán información sobre:

- **Inicio:** programas y aplicaciones a cargar en el inicio de sesión.
- **Datos de programa:** datos específicos de las aplicaciones (como idioma, perfil de ejecución, etc.).
- **Cookies:** información y preferencias de navegación del usuario.
- **Escritorio:** qué elementos mostrar en el escritorio (archivos, accesos directos, carpetas, fondo de pantalla, etc.).
- **Favoritos:** listado de direcciones favoritas de Internet.
- **Configuración local:** archivos temporales, historial de ejecución, datos de aplicaciones.
- **“Mis documentos”:** documentos y subcarpetas propios del usuario (en entorno Windows se llama así, pero en otros sistemas podrá tener otro nombre).
- **Documentos recientes:** memoria de acceso a los documentos utilizados recientemente.

- **Entorno de red:** accesos directos a elementos de red.
- **Impresoras:** configuración de cuáles usa el usuario entre todas las disponibles en la red.
- **Plantillas:** repositorios de recursos para el trabajo del usuario.

## Sistema de carpetas y directorios

En general, para cada usuario existirá en el sistema una serie de carpetas y directorios directamente relacionados con la información de su cuenta y su perfil de usuario. Aunque puede ser similar en una misma familia de sistemas operativos (p. ej. diferentes versiones de Windows o distribuciones de Linux/Unix), siempre conviene revisar cómo es en particular en el sistema con el que estemos trabajando.

### WINDOWS

La primera vez que se inicia la sesión de usuario se crea una carpeta para su perfil y se copia en ella el contenido de la carpeta del perfil por defecto, “Default User” (esto depende de la versión), que junto con las configuraciones de los grupos comunes (carpetas “allusers” o “acceso publico”) se usan para crear su perfil de escritorio. Cuando el usuario termina su sesión todos los cambios realizados se guardan sobre su perfil, manteniendo sin modificar el perfil por defecto.

Según el tipo de perfil los cambios se almacenan en:

- **Perfil local:** se mantienen en un directorio predeterminado o en una ruta del perfil (propiedades del usuario). Los directorios puede ser (según versiones):

C:\Users\nombreusuario\ntuser.dat

C:\Documents and Settings\nombreusuario\ntuser.dat

- **Perfil de usuario móvil:** el perfil se almacena en el fichero “ntuser.dat”, pero estará ubicado en el servidor, en una ruta especificada por el administrador. Ejemplo:  
\\nombresservidor\carpetaperfiles\nombreusuario\
- **Perfil de usuario obligatorio:** igual que el anterior, pero se mantiene sin modificaciones del usuario. El fichero “ntuser.dat” pasará a llamarse “ntuser.man”.

## UNIX - LINUX

Se utilizan archivos de texto en los que la información se almacena línea a línea (una para cada usuario o grupo), y dentro de cada línea se especifican varios campos separados por ":". Los ficheros son:

- / etc / **passwd**: mantiene información sobre la **cuenta de usuario y la contraseña**, junto con la mayor parte de la información sobre cuentas en el sistema Unix.
- / etc / **shadow**: Tiene la **contraseña cifrada** de la cuenta correspondiente, aunque no está soportado por todos los sistemas.
- / etc / **group**: **Información de los grupos** para cada cuenta de usuario.
- / etc / **gshadow**: **Información de seguridad** de los grupos de la cuenta.

Existen otros ficheros como pueden ser los de “*scripts*” de inicio o final de sesión, y que se guardan en el directorio **/etc/skel/**, por ejemplo para inicializar variables del sistema, de aplicaciones, etc. Te lo mostramos en un servidor Linux:

```
[pqwert055@server-1 skel]$pwd  
/etc/skel  
[pqwert055@server-1 skel]$ls -al  
total 32  
drwxr-xr-x. 2 root root 4096 Jan 13 03:08 .  
drwxr-xr-x. 82 root root 12288 Apr  3 21:10 ..  
-rw-r--r--. 1 root root     0 May 31  2017 .bash_history  
-rw-r--r--. 1 root root    18 Dec  7  2016 .bash_logout  
-rw-r--r--. 1 root root   193 Dec  7  2016 .bash_profile  
-rw-r--r--. 1 root root   231 Dec  7  2016 .bashrc  
-rw-r--r--. 1 root root  658 Aug  2  2017 .zshrc  
[pqwert055@server-1 skel]$
```



Como verás cuando practiques sobre el sistema, gran parte de estos directorios están ocultos. Para verlos podemos usar varios métodos:

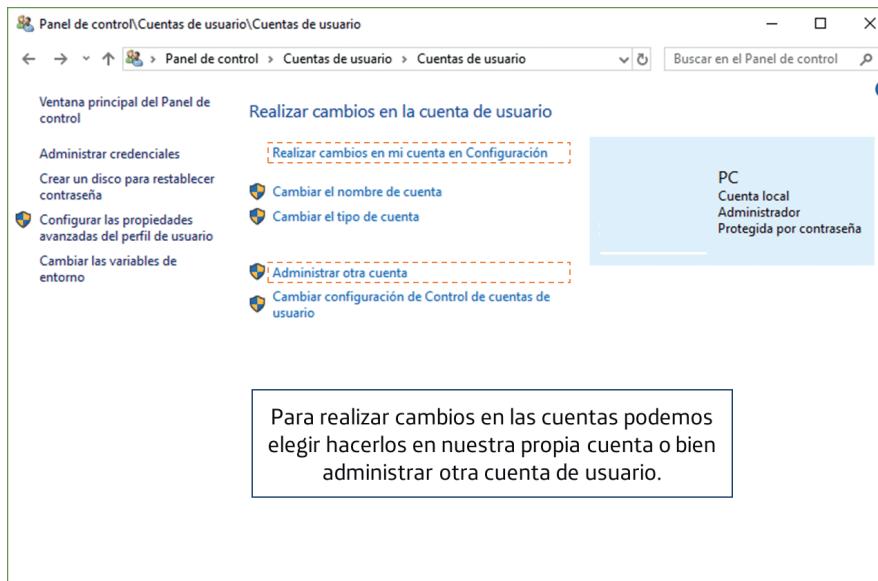
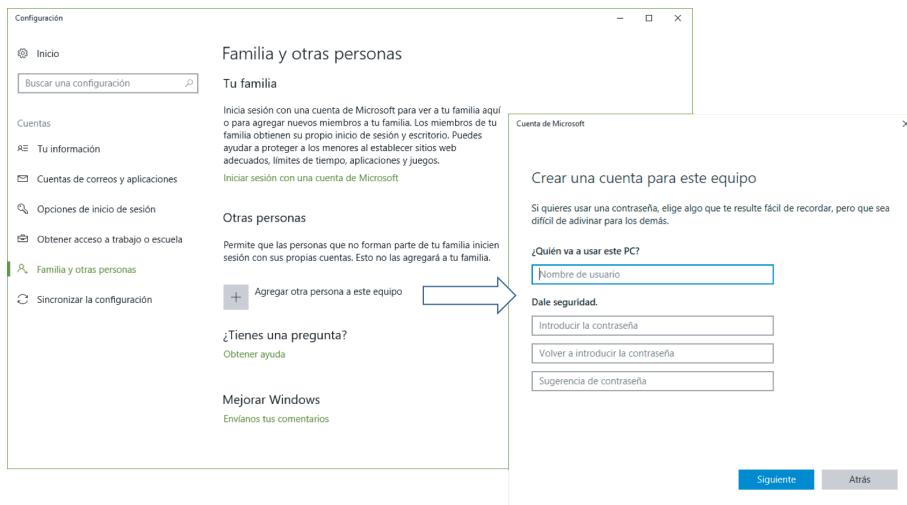
**En Windows:** en el explorador de archivos eligiendo la opción "**Vista - Elementos ocultos**", y en la consola de comandos con "**dir /A**" vemos todos los archivos y sus atributos.

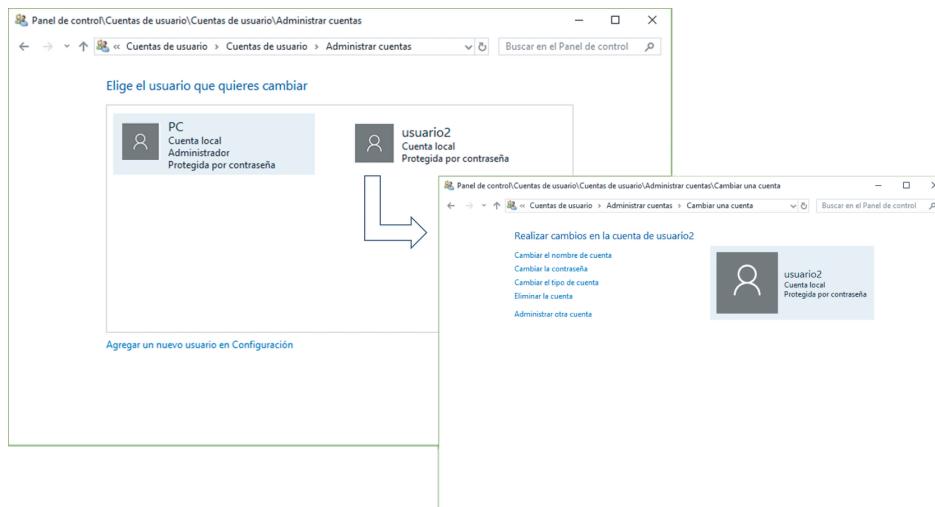
**En Linux:** con el comando "**ls -al**" en el terminal o pulsando "**ctrl+h**" en el explorador de archivos.

# Información del perfil de usuario en Windows

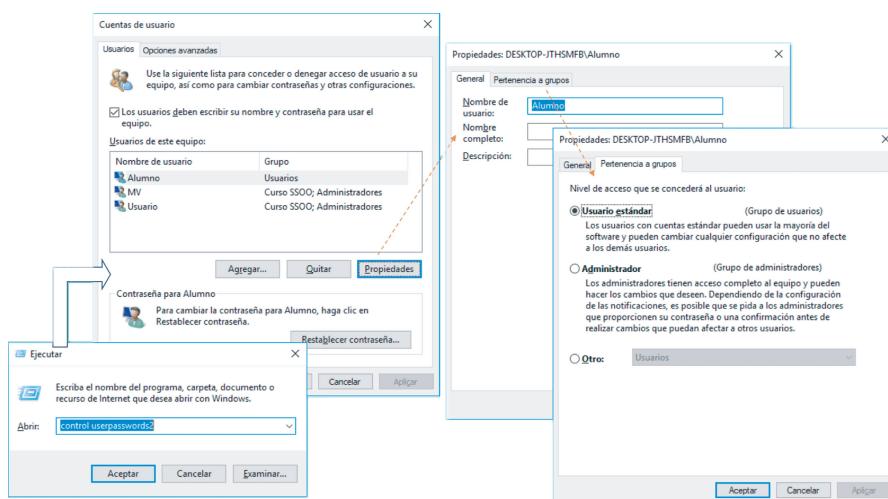
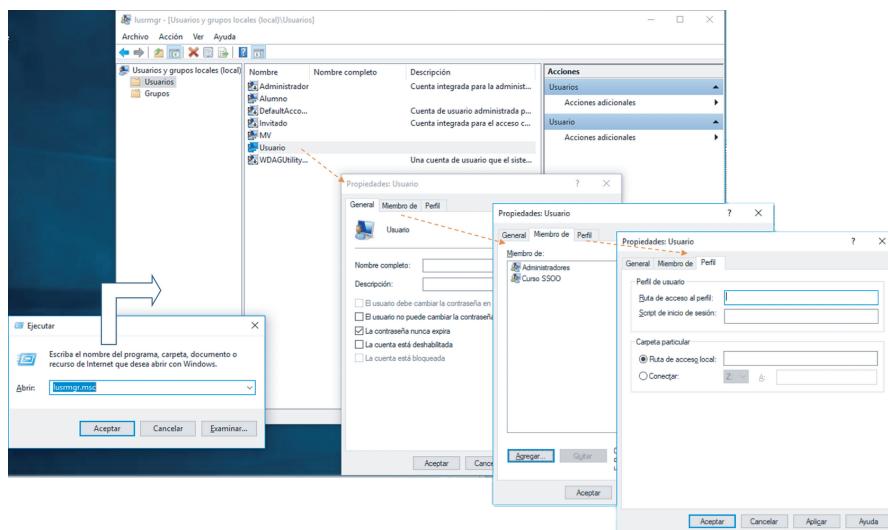
Para ver la información del perfil de un usuario las opciones pueden variar un poco dependiendo de la versión del sistema.

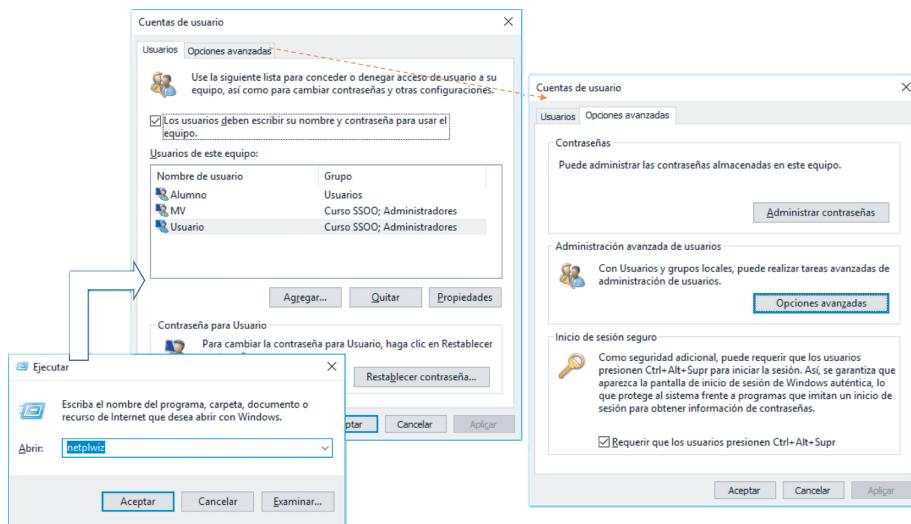
Si estamos en **Windows 10 Home** tenemos menos recursos a nuestra disposición y lo mejor es trabajar con los usuarios a través del "gestor de cuentas de usuario" del "panel de control".





Si estamos en alguna de las versiones "PRO" de Windows el sistema nos permite ejecutar las herramientas "lusrmgr.msc", "control userpasswords2" y "netplwiz", que nos dan acceso a los datos del usuario y su pertenencia a grupos.





## El archivo de registro Windows y el fichero Ntuser.dat

Desde hace varias generaciones, Windows incorpora un sistema mediante el cual almacena en una base de datos jerarquizada propia una gran cantidad de información sobre el funcionamiento del sistema; desde información de bajo nivel sobre las variables de entorno, hasta las aplicaciones instaladas, los ajustes de configuración, servicios, perfiles de usuario, etc. Esta base de datos se conoce como "**Registro de Windows**" y está compuesta por dos elementos básicos:

- **claves**: son como subcarpetas dentro del registro que contienen información de un determinado tipo.
- **valores**: son pares de nombres y datos que se encuentran dentro de las claves.

Aunque pueden variar de un sistema a otro, algunas de las claves que podemos encontrar dentro del registro son, por ejemplo:

- HKEY\_LOCAL\_MACHINE (HKLM) - información de configuración del equipo local.
- HKEY\_CLASSES\_ROOT (HKCR) - información sobre aplicaciones registradas.
- HKEY\_CURRENT\_USER (HKCU) - configuración del usuario que tiene la sesión activa.
- HKEY\_PERFORMANCE\_DATA - información de ejecución y datos de rendimiento.

El registro como tal puede además importar datos de varios ficheros que tienen el formato ".DAT" (ficheros de registro), como por ejemplo "NTUser.dat".

El registro de Windows puede editarse con la herramienta "regedit", pero puede resultar complejo y arriesgado si no se tiene un conocimiento avanzado del sistema.



En Windows cada perfil de usuario tiene un archivo **Ntuser.dat** con información sobre el propio perfil. Un perfil de usuario estará formado por archivos personales y parámetros de las preferencias del propio usuario, la configuración de su escritorio, historial de navegación, etc.

El archivo **Ntuser.dat** es **un archivo oculto** que contiene la configuración del registro de su cuenta individual. Además existen otros archivos relacionados con él, con información histórica que va creando el propio sistema.

La visualización del archivo **NTUser.dat** en los sistemas Windows no es sencilla, pues se encuentra codificado y no es legible en formato texto. Además, el sistema se protege a sí mismo y no dejará acceder al archivo del usuario que tiene activa la sesión. Si queremos ver su información deberemos acceder desde otro usuario.

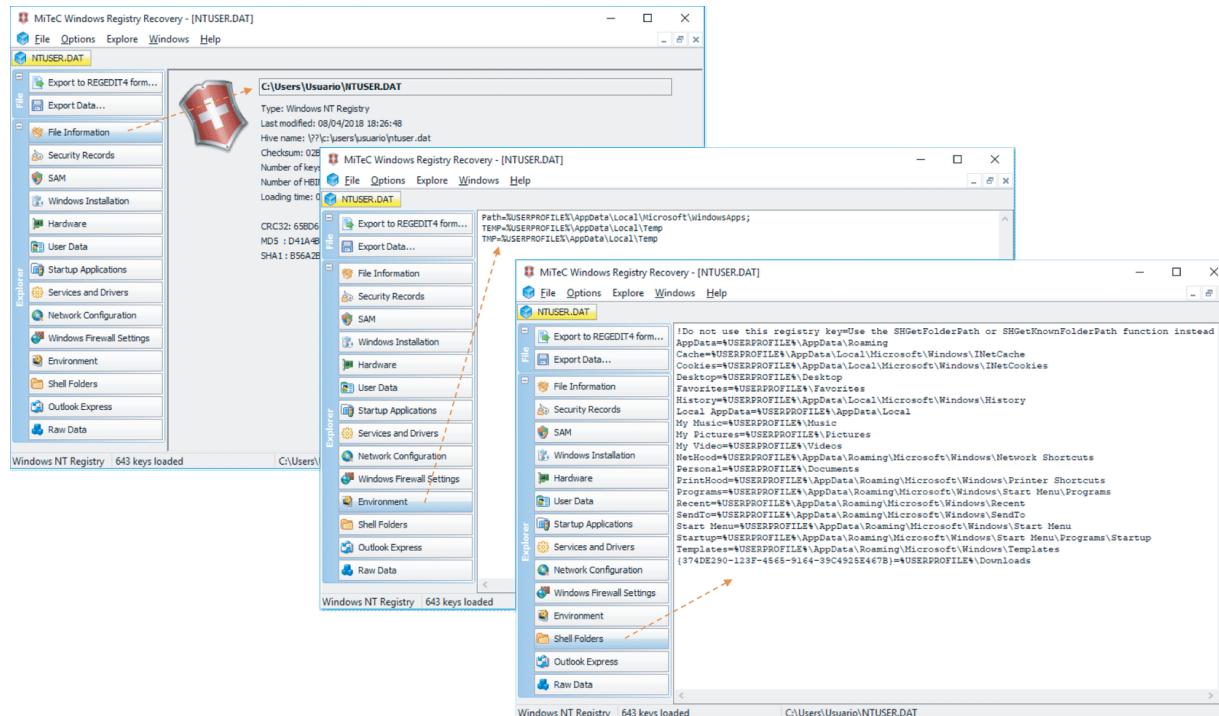
```
C:\Users\MV>dir /A *NTUser*
El volumen en la unidad C no tiene etiqueta.
El n mero de serie del volumen es: CC21-CB47

Directorio de C:\Users\MV

04/04/2018 13:08    1.310.720 NTUSER.DAT
19/03/2018 22:56    387.072 ntuser.dat.LOG1
19/03/2018 22:56    387.072 ntuser.dat.LOG2
20/03/2018 00:25    65.536 NTUSER.DAT{df92b72d-2bb1-11e8-9b9e-000c29b44dc9}.TM.blf
20/03/2018 00:25    524.288 NTUSER.DAT{df92b72d-2bb1-11e8-9b9e-000c29b44dc9}.TMContainer000000000000000000000001.regtrans-ms
20/03/2018 00:25    524.288 NTUSER.DAT{df92b72d-2bb1-11e8-9b9e-000c29b44dc9}.TMContainer000000000000000000000002.regtrans-ms
19/03/2018 22:56          20 ntuser.ini
                           7 archivos      3.198.996 bytes
                           0 dirs   49.484.038.144 bytes libres

C:\Users\MV>
```

Para poder verlo, además de visualizar los archivos ocultos de la carpeta, podemos utilizar una aplicación de terceros, como puede ser "Mitec Windows Registry Recovery" (<http://www.mitec.cz/wrr.html>), que mostramos a continuación:



---

**Te recomendamos que, al menos al principio, gestiones la información del perfil de los usuarios a través de las propias aplicaciones de gestión de cuentas que hemos visto antes.**

# Información de perfil de usuario en Linux

Vamos a echar un vistazo a algunos de los ficheros que guardan información sobre el perfil de los usuarios. Te recomendamos que los explores en tu máquina virtual.

## El fichero /etc/passwd

Puedes verlo mediante la línea de comando o desde el editor gráfico. Evidentemente, tanto este como los demás ficheros de configuración, no conviene modificarlos directamente, sino modificar las características de los usuarios a través de los comandos del sistema.

passwd [Solo lectura] (/etc) - gedit

Guardar

```
root:x:0:0::/root:/bin/bash
daemon:x:1:1::/usr/sbin/nologin
bin:x:2:2::/bin:/usr/sbin/nologin
sys:x:3:3::/dev:/usr/sbin/nologin
sync:x:4:65534::sync:/bin:/sync
games:x:5:60::/usr/games:/usr/sbin/nologin
man:x:6:12::/var/cash:/usr/sbin/nologin
lp:x:7:7::lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8::mail:/var/mail:/usr/sbin/nologin
news:x:9:9::news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10::uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13::proxy:/bin:/usr/sbin/nologin
www-data:x:33:33::www-data:/var/www:/usr/sbin/nologin
backup:x:
list:x:3
irc:x:39
gnats:x
nobody:x
systemd-sustained-updates:x
otrousuario@ubuntu:~
```

Podemos visualizar el fichero /etc/passwd desde el administrador de archivos de la interfaz gráfica de Ubuntu, haciendo click sobre él se abrirá con el editor gráfico "gedit", o bien a través de comando desde el terminal.

En él cada línea hace referencia a un usuario y nos da información sobre él: <nombre>,<password><uid>,<gid>,<descripción opcional> <carpeta>,<shell>.

otrousuario@ubuntu:~\$ cat /etc/passwd | grep "otrousuario"
otrousuario:x:1002:1007:otrousuario,,,:/home/otrousuario:/bin/bash

Nombre de usuario  
Identificador del grupo principal del usuario (GID). Puede pertenecer a otros grupos también.  
Identificador de usuario (UID) entre 0 (root) y 65535. Algunos están reservados, por ej. root = 0 siempre.  
Descripción (opcional)  
Carpeta de inicio del usuario  
Shell (intérprete de mandatos) de inicio asignada al usuario  
"x" indica que el password está almacenado en /etc/shadow, en el caso de ser una "!" es que el usuario está bloqueado. Si tiene "!!" es que no tiene.

## El fichero /etc/shadow

Este archivo, además de la contraseña cifrada, contiene información sobre la caducidad de las cuentas de los usuarios. Solamente puede ser leído por el superusuario (root). Te lo mostramos.

```

otrousuario@ubuntu:~$ cat /etc/shadow
otrousuario:$6SeWATT9g9Sg94y17/Bgy.3dHJ1yVPPhZactVYjpkvB78xu9jFBTU6MpknGUv7/095ZkaHBZgSGFFc0tjE3nuBGpbtBHmD/:17625:0:99999:7::17652:
otrousuario:#
root@ubuntu:~$ exit
exit
otrousuario@ubuntu:~$
```

Días transcurridos desde 1-1-1970 donde el password fue cambiado por última vez.

Cantidad de días (mínimo) que tiene que haber entre cambios de contraseña

Días (max) de validez de la cuenta

Aviso (en días) antes de caducar la contraseña

Fecha de caducidad de la cuenta (en días desde 1-1-1970)

Días hasta que se deshabilita la cuenta después de que caduque la contraseña.

## El fichero /etc/group

Nos da información sobre los grupos, de forma similar a la información sobre usuarios que nos da /etc/passwd. Veámoslo:

```

group [Solo lectura] (/etc) - gedit
Abrir ▾ Guardar
Sambarshare:x:128.usuarios
guest-26wrmix:x:999:
direccion:x:1002:
comercial:x:1003:
rrhh:x:1004:
operaciones:x:1005:
contabilidad:x:1006:
otrousuario:x:1007:
administrador:x:1008:
guest-lyqwmb:x:998:
pepe:x:1001:
```

```

otrousuario@ubuntu:~$ cat /etc/group | grep "contabilidad"
contabilidad:x:1006:pepe,otrousuario
otrousuario@ubuntu:~$ cat /etc/group | grep "operaciones"
operaciones:x:1005:
otrousuario@ubuntu:~$ cat /etc/group | grep "administrador"
administrador:x:1008:
otrousuario@ubuntu:~$ cat /etc/group | grep "otrousuario"
contabilidad:x:1006:pepe,otrousuario
otrousuario:x:1007:
otrousuario@ubuntu:~$ cat /etc/group | grep "root"
root:x:0:
otrousuario@ubuntu:~$
```

Nombre de grupo

x = contraseña en fichero /etc/gshadow

Miembros del grupo

GID = identificador del grupo

## El fichero /etc/gshadow

Contiene la información oculta sobre los grupos de usuarios y solamente puede ser visualizado por el superusuario (root), aunque sí podemos visualizar su descripción con "man gshadow".

Te lo mostramos.

```
root@ubuntu:/home/otrousuario
otrousuario@ubuntu:~$ cat /etc/gshadow
cat: /etc/gshadow: Permiso denegado
otrousuario@ubuntu:~$ su
Contraseña:
root@ubuntu:/home/otrousuario# cat /etc/gshadow | grep "otrousuario"
contabilidad:::pepe,otrousuario
otrousuario@ubuntu:~$ cat /etc/gshadow | grep "contabilidad"
contabilidad:::pepe,otrousuario
root@ubuntu:/home/otrousuario#
root@ubuntu:/home/otrousuario# cat /etc/gshadow | grep "administrador"
administrador:::
root@ubuntu:/home/otrousuario# cat /etc/gshadow | grep "root"
root:::
root@ubuntu:/home/otrousuario#
```

Miembros del grupo

Nombre del grupo

Ambos significan que la contraseña esta encriptada y no se puede acceder al grupo usando una clave. Los miembros del grupo no necesitan contraseña

```
otrousuario@ubuntu:~$ man gshadow
```

File Formats and Conversions GSHADOW(5)

```
NAME      gshadow - shadowed group file
DESCRIPTION /etc/gshadow contains the shadowed information for group accounts.
This file must not be readable by regular users if password security is to be maintained.
Each line of this file contains the following colon-separated fields:
group name
It must be a valid group name, which exist on the system.
encrypted password
Refer to crypt(3) for details on how this string is interpreted.
If the password field contains some string that is not a valid result of crypt(3), for instance ! or *, users will not be able to use a unix password to access the group (but group members do not need the password).
The password is used when an user who is not a member of the group wants to gain the permissions of this group (see newgrp(1)).
Manual page gshadow(5) line 1 (press h for help or q to quit)
```

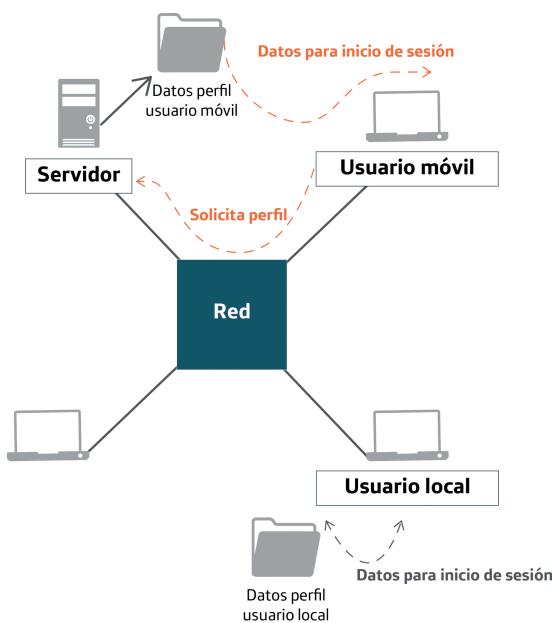
**i** Existen más ficheros que se usan para configurar el perfil del usuario, como por ejemplo ".profile", ".bashrc", "~/.config/user-dirs.dirs", etc., que irás viendo con el tiempo y conforme vayas profundizando en tu conocimiento de Linux.

# Cambiar la ruta por defecto del usuario

En algunos casos podemos querer **cambiar la ruta por defecto** en la que se almacena la información del perfil de usuario. Normalmente esta definición se hará en el momento de creación de la cuenta, pero puede también realizarse después.

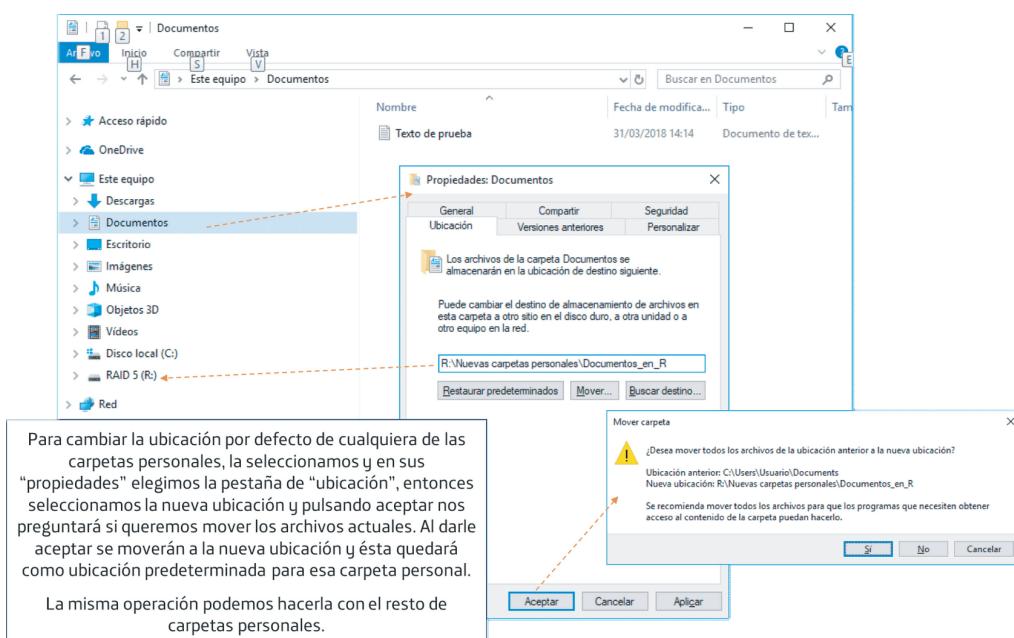
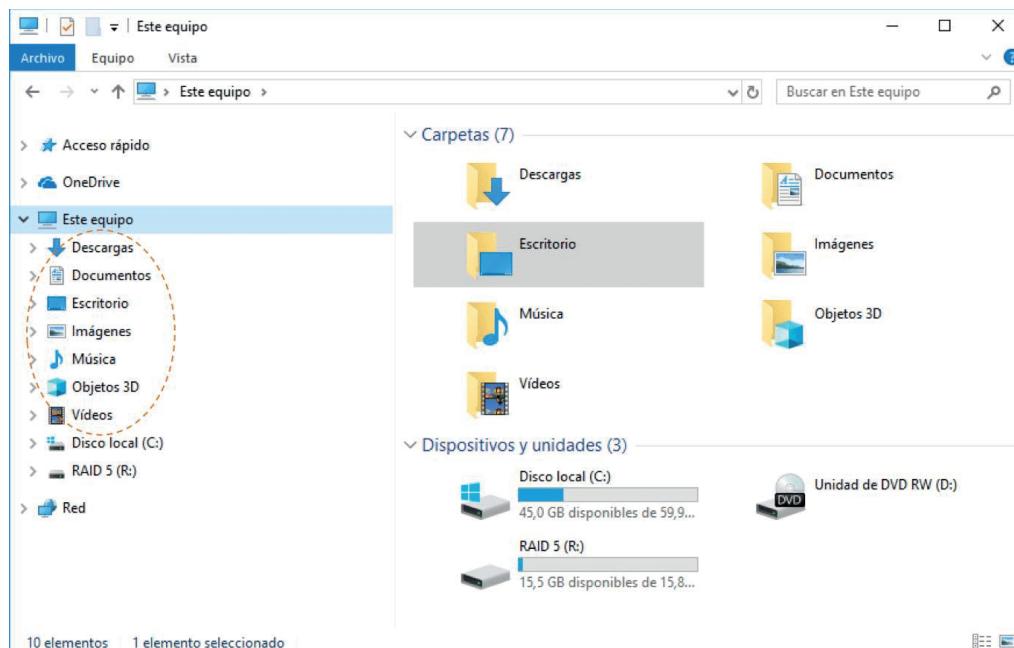
Sobre todo cuando estemos utilizando **perfiles móviles**, debemos disponer de una **carpeta compartida donde almacenar los perfiles** de los usuarios que van a acceder desde algún equipo de la red, y luego configurar cada usuario para que acceda a ella para cargar su perfil "personal", independientemente del equipo en el que inicie la sesión.

Si estamos trabajando **a nivel local** la operación puede ser más sencilla, ya que cada sistema operativo nos ofrecerá normalmente una forma de configurar la **ruta de acceso a las carpetas del perfil**, generalmente a través de la interfaz gráfica o bien por línea de comandos. Por ejemplo, si estamos trabajando en Linux, con el comando “`usermod -d nombreusuario`” podemos modificar el directorio de trabajo del usuario.



## Cambiar la ruta por defecto de las carpetas personales en Windows

En Windows podemos cambiar la ruta por defecto de las carpetas personales fácilmente desde el propio administrador de archivos. Simplemente seleccionamos la carpeta a la que deseamos cambiar la ruta por defecto y con el botón derecho elegimos ver sus propiedades, después la pestaña de "ubicación" y ahí podemos seleccionar la nueva ruta de la carpeta, que puede ser local o bien un disco en la red. Te lo mostramos en las siguientes pantallas:

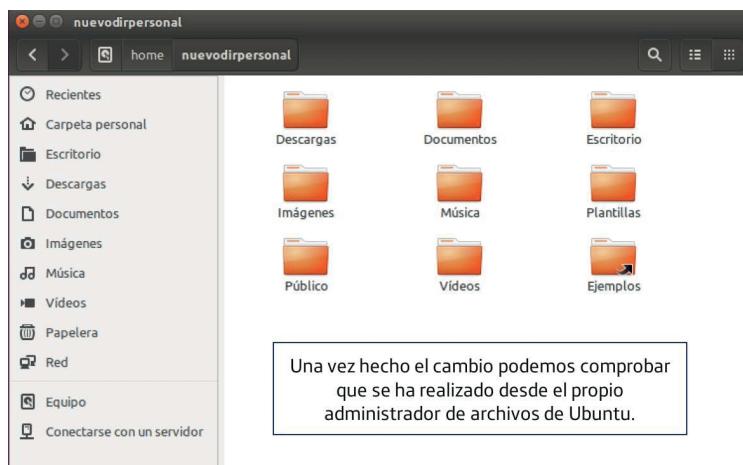
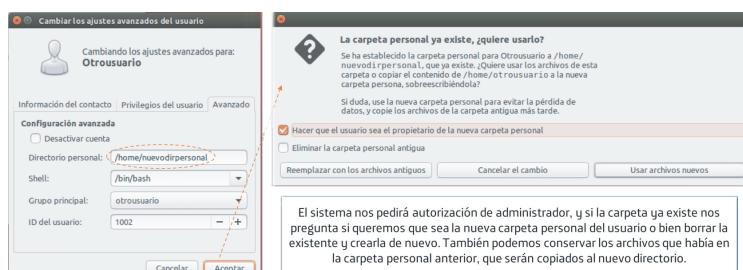


## Cambiar la ruta por defecto de las carpetas personales en Linux

Si estamos en Linux Ubuntu podemos usar la interfaz gráfica para modificar la ruta por defecto de un usuario, pero siempre desde un usuario con permisos de administración sobre otro usuario del sistema, y que no tenga sesión activa en ese momento.

Para hacerlo accedemos a la ventana de "Ajustes de usuarios" y en la opción de "Ajustes avanzados" podemos configurar un nuevo directorio. Si es una carpeta que ya existe en el sistema nos preguntará si deseamos que el usuario se convierta en el "dueño" de esa carpeta y si queremos mover los ficheros ya existentes al nuevo destino.

Podemos verlo en la siguiente secuencia de imágenes:



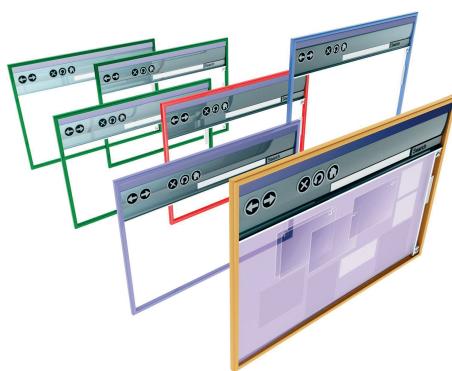
# Acceso a recursos y permisos locales

En un sistema multiusuario, cada uno de ellos tiene una serie de derechos y privilegios que le permiten realizar ciertas acciones (ejecutar aplicaciones, hacer copias de seguridad, realizar ciertas labores de administración, etc.), así como una serie de permisos.

Estos permisos suelen ser reglas asociadas a un elemento u objeto, generalmente **archivos** y **directorios/carpetas**.

Un **objeto** (en este contexto) es cualquier elemento que puede asegurarse y protegerse mediante permisos. Puede ser un fichero, un directorio, una clave de registro, un proceso, etc.

Los **permisos** son entonces propiedades de los elementos (p. ej. ficheros) que pueden asignarse cuando se crea ese objeto y modificarse después. Por otro lado, los **derechos y privilegios van asociados a las cuentas de usuario**, ya sean estas locales (definidas en un equipo concreto y el usuario no puede usarlas en otro) o globales (el usuario puede usar cualquier equipo y hace “*login*” contra un servidor de red o de dominio).



## La herencia

En muchos casos el acceso a los recursos tiene la propiedad de “**herencia**”, es decir, que los permisos se pueden aplicar a otros objetos contenidos o “hijos” del elemento inicial. Por ejemplo las propiedades de solo lectura de un directorio se transmiten a los subdirectorios que dependen de él.

Puede haber entonces ocasiones en las que un permiso sea contrario a otro, y en ese caso el orden de interpretación de los permisos debería ser:

- 1 Accesos denegados explícitamente (“**deny only**”).
- 2 Accesos autorizados explícitamente (“**allow only**”).
- 3 Accesos denegados por herencia.
- 4 Accesos autorizados por herencia.

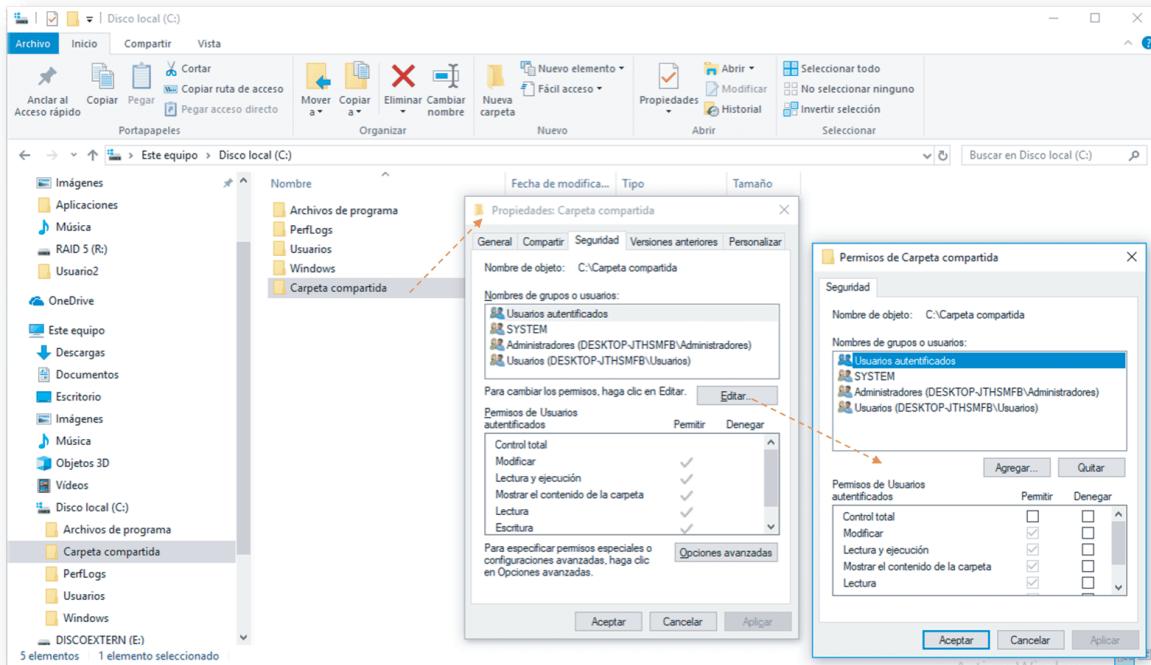
Por ejemplo, si un directorio tiene permiso de escritura, en principio todos los subdirectorios por debajo de él también lo tendrán, pero si explícitamente se pone alguno de ellos como “solo lectura”, eso debe “mandar” sobre el permiso de escritura heredado.

# Permisos en Windows

La gestión de permisos en Windows a nivel local no es demasiado compleja y está bastante accesible desde el entorno gráfico.

## Permisos en Windows

En el caso de Windows, la modificación de permisos a nivel local puede hacerse a través del administrador de archivos seleccionando la unidad (por ejemplo C:) y con el botón derecho del ratón seleccionando “Propiedades”, y luego en la pestaña “Seguridad” elegir la opción de “Editar”. Por supuesto, también podemos modificar los permisos por línea de comandos.



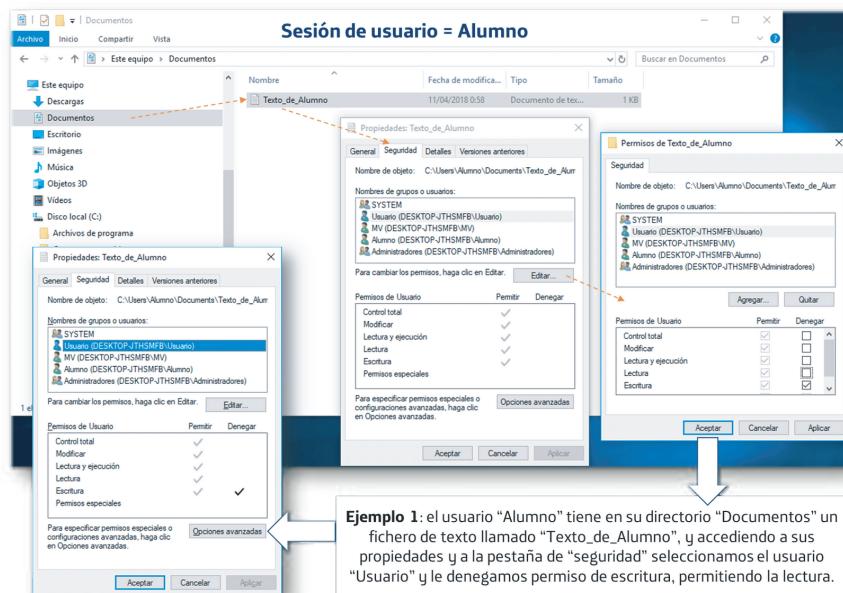
Recuerda que cada sistema operativo creará sobre el sistema de ficheros (por ejemplo NTFS) una serie de permisos por defecto para determinadas carpetas y directorios; debemos consultar la información detallada del S.O. si queremos conocerlos.

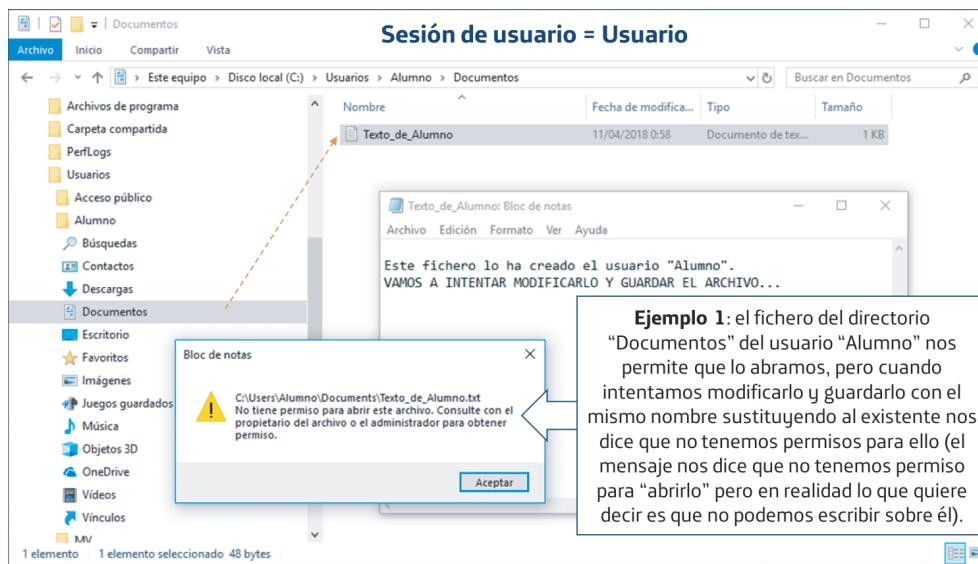
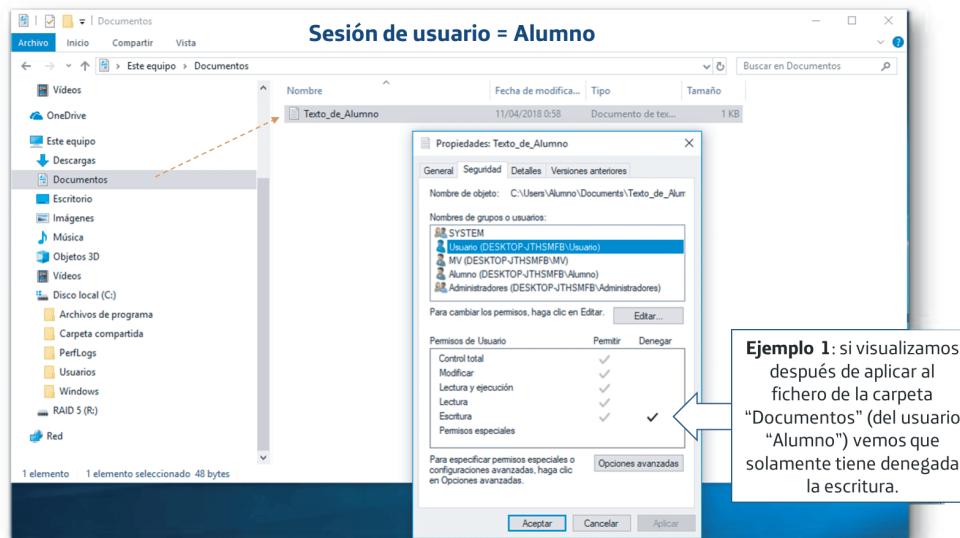
Lo más fácil para modificar los permisos de un fichero en Windows es hacerlo seleccionándolo en el administrador de archivos y con el botón derecho del ratón activando la ventana de "Propiedades".

## Ejemplos de modificación de permisos en Windows

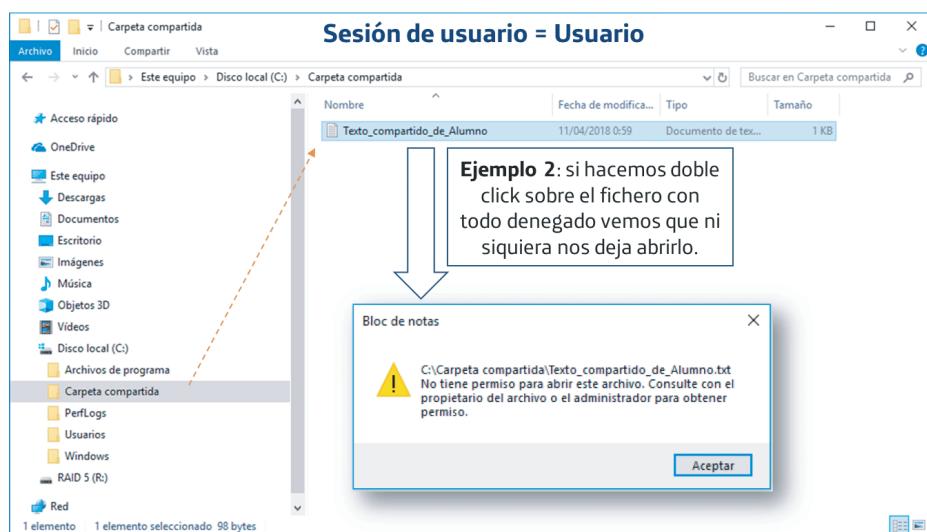
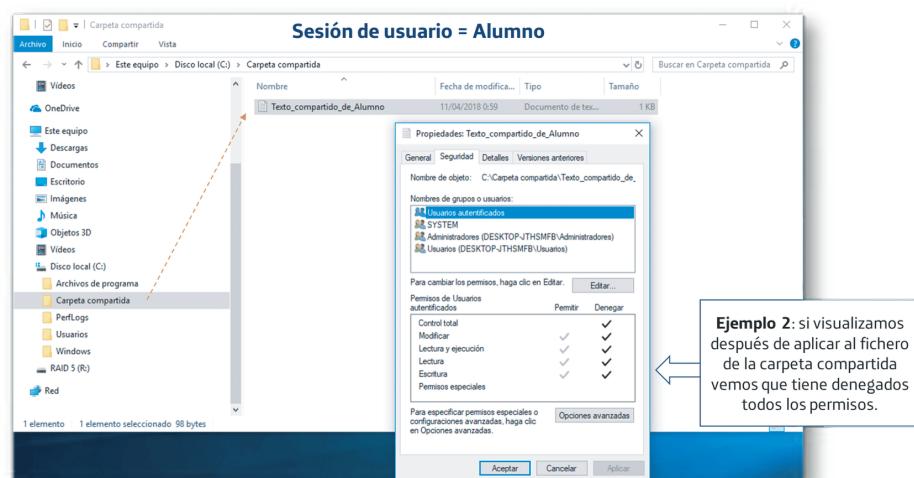
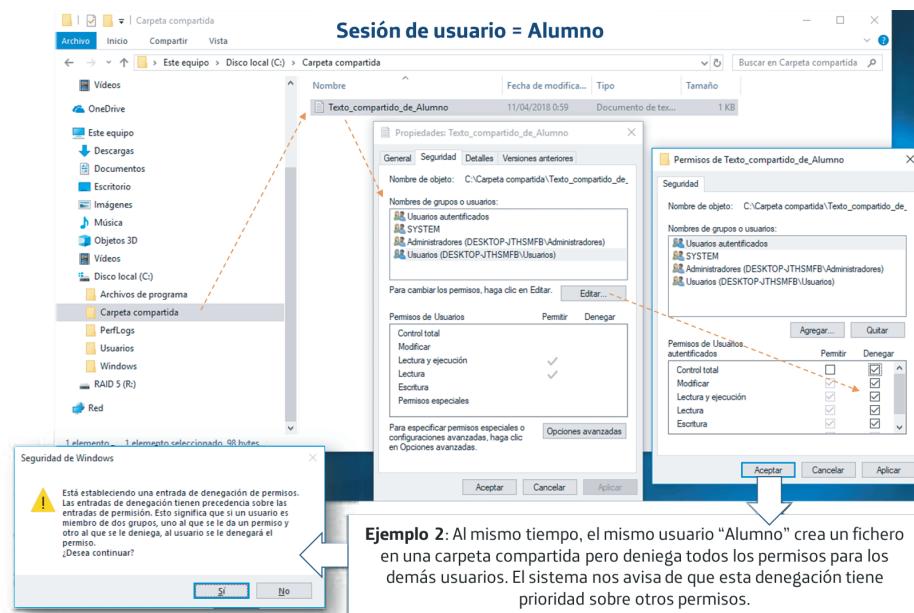
Veamos primero un caso en el cual permitimos a otro usuario acceder a un fichero del directorio personal "Documentos" de un usuario ("Alumno"), de forma que pueda visualizar el contenido pero no modificarlo. Ten en cuenta que el segundo usuario ("Usuario") deberá tener permisos de acceso al directorio personal del primero ("Alumno"), y esto es independiente de lo que luego le permitamos hacer con el fichero (p. ej. "Texto\_de\_alumno").

Te lo mostramos en una secuencia de pantallas:





Ahora veamos que el mismo usuario ("Alumno") crea un fichero en una carpeta compartida del sistema (fuera de su espacio personal) pero le deniega todos los permisos, incluso el de lectura, y luego desde otro usuario intentamos abrir el archivo y no nos lo permitirá.



# Permisos en Linux

---

Ya conocemos un poco del sistema de permisos que utiliza Linux para permitir acciones sobre ficheros y directorios, pero ahora vamos a profundizar un poco más en su utilización.

## Linux

En sistemas Linux/Unix cada "objeto", archivo y carpeta también tiene un **propietario** y está **asignado a un grupo**, normalmente al que pertenece el propietario. En general los archivos/carpetas tienen los tres conjuntos de permisos "**rwx**" que, aunque ya los conoces, vamos a ver con algo más de detalle.

Estos permisos pueden ser:

- Permiso de lectura / acceso.
- Permiso de escritura / modificación.
- Permiso de ejecución.

A su vez estos tres tipos de permisos pueden aplicarse a:

- El propietario (usuario).
- Los miembros del mismo grupo al que pertenece el archivo.
- El resto de usuarios.

Veamos estos permisos con algo más de detalle.

### Permisos básicos en Unix/Linux

Son los que ya conoces, pero vamos a repasarlos y ver las diferencias entre archivos o directorios:

Permisos básicos en Unix/Linux	Lectura/Listado "r"	Escritura "w"	Ejecución/Acceso "x"
Directorio	Se puede listar el contenido del directorio	Se pueden crear ficheros dentro del directorio	Se puede entrar en el directorio
Archivo	Se puede leer el contenido del fichero	Se puede modificar el contenido del fichero	Se puede ejecutar el archivo como un programa  ↑ <i>¡Siempre que sea un ejecutable o un script de metadatos, claro!</i>

### Permisos especiales en Unix/Linux

Estos pueden ser un poco más complicados, pero los veremos con un ejemplo más adelante.

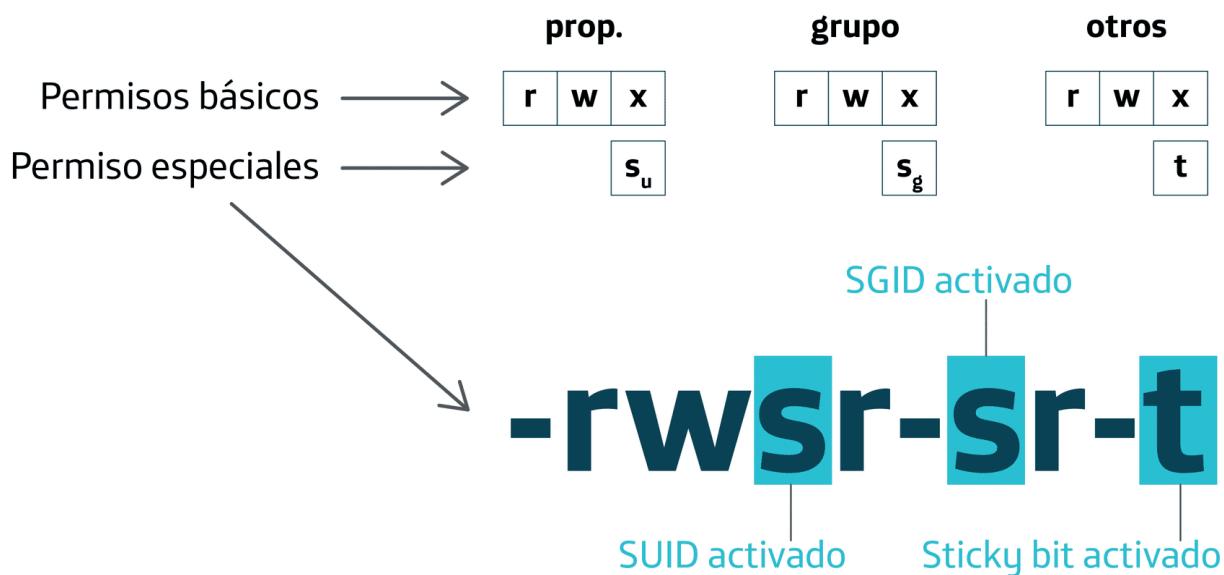
Permisos especiales en Unix/Linux	Set-uid "su"	Set-gid "sg"	Sticky "t"
Directorio	(No utilizado)	Se crean nuevos archivos con el grupo de la carpeta "padre"	Aunque todos puedan crear archivos en el directorio, solo el propietario del archivo (y root) pueden eliminarlo
Archivo	Ejecución con privilegios del propietario del programa	Ejecución con privilegios del grupo del propietario del programa	(No utilizado)

### Notación simbólica en Unix/Linux

Recuerda que a la hora de asignar permisos a un fichero o directorio cada clase de permisos se representa por **tres caracteres**: los tres primeros relativos al **usuario**, los tres siguientes relativos al **grupo** y los otros tres al **resto**. En cada posición aparecerá una letra si tiene asignado ese permiso, y un signo “-” si no lo tiene. El primer carácter de la izquierda representa el tipo de archivo/directorio/etc. Por ejemplo:

- "**-rwxr-xr-x**": archivo normal con todos los permisos para su propietario, pero solo permisos de lectura y ejecución para el grupo de usuarios del archivo y el resto de los usuarios. Ningún usuario, salvo el propietario, puede modificar los contenidos del archivo.
- "**dr-x-----**": directorio que tiene permisos de lectura y ejecución solamente para su propietario.
- "**crw-rw-r--**": archivo especial de caracteres con permisos de lectura y escritura para propietario y grupo, y solo permiso de lectura para el resto de los usuarios.

Los permisos especiales (SUID, SGID y Sticky bit) se añadirán a estos como vemos a continuación:



### Notación octal en Unix/Linux de permisos básicos y especiales

Como sabes, la notación de los permisos en Unix/Linux para un archivo/directorio a menudo también se expresa por una secuencia de números sustituyendo a cada uno de los tres grupos de letras (usuario, grupo, resto).

La forma de hacerlo es haciendo la conversión a octal (base 8) de cada grupo de tres signos, que tendrán en cada posición un “0” si no tiene el permiso o un “1” si lo tiene.

Para comprenderlo mejor veamos algunos ejemplos (sin contar con el signo de tipo de archivo):

- "**-rwxr-xr-x**": equivaldría a “111 101 101”, que en octal sería “755” o “0755” si lo expresamos con cuatro dígitos (esto nos servirá para los permisos especiales).
- "**-rw-rw-r--**": equivale a “110 110 100”, que en octal sería “664” o “0664”.
- "**-r-x-----**": equivale a “101 000 000”, que en octal sería “500” o “0500”.

**IMPORTANTE**

**¿Y cómo se representan los permisos especiales?** Añadiendo un cuarto dígito que precede a los anteriores, y que se calcula con la suma de los pesos de [setuid (4) + setgid (2) + sticky (1)] cuando están presentes.

Por ejemplo:

"**-rwsr-sr-x**": como tiene "setuid" y "setgid" ( $4+2=6$ ) se representaría como 6745.

Como vemos arriba, cuando no hay permisos especiales se pone un cero al principio o se usa la notación de tres dígitos.

**Ejemplo de notación de permisos especiales**

Los permisos especiales se indican cambiando uno de los tres permisos de ejecución según se indica en la figura siguiente.

- “**-rwsr-sr-x**”: por ejemplo, este archivo tiene permisos de lectura, escritura y ejecución para el usuario, de lectura para el grupo, y de lectura y ejecución para el resto. Además tiene los permisos de “setuid” y “setgid” asignados al grupo.

Lo resumimos en esta tabla y lo veremos en ejemplos más adelante:



Permiso	Clase	Ejecutable	No ejecutable
Set User ID (setuid)	Usuario	s	S
Set Group ID (setgid)	Grupo	s	S
Sticky	Otros	t	T

Si lo tiene, es que además del especial tiene permiso de ejecución



Si lo tiene, es que además del especial tiene NO permiso de ejecución



# Cómo cambiar los permisos de un archivo en Linux

Lo primero que debes recordar es que:

*"En Linux todo es un fichero"*

Por lo tanto utilizaremos el mismo método para gestionar los permisos de archivos y directorios.

## Cómo usar el comando "chmod"

El comando “chmod” nos permite gestionar los permisos de ficheros y directorios en Linux. Podemos usarlo utilizando la notación simbólica o la notación octal. El comando tiene más opciones de las que veremos en estos ejemplos, y te recomendamos que las visualices con “man chmod” desde un terminal de tu S.O. Linux Ubuntu.

La forma general del comando es: “**chmod [opciones] modo[,modo] fichero**”.

Para aplicarlo usando la notación simbólica debemos tener en cuenta que nos referimos a los grupos de permisos como:

- **u**: usuario dueño del fichero.
- **g**: grupo de usuarios del dueño del fichero.
- **o**: todos los otros usuarios.
- **a**: todos los tipos de usuario (dueño, grupo y otros).

Y para asignar uno u otro permiso:

- **r**: se refiere a los permisos de lectura.
- **w**: se refiere a los permisos de escritura.
- **x**: se refiere a los permisos de ejecución.

También podemos usar la notación octal para establecer los permisos. Veremos ambas posibilidades a través de unos ejemplos.



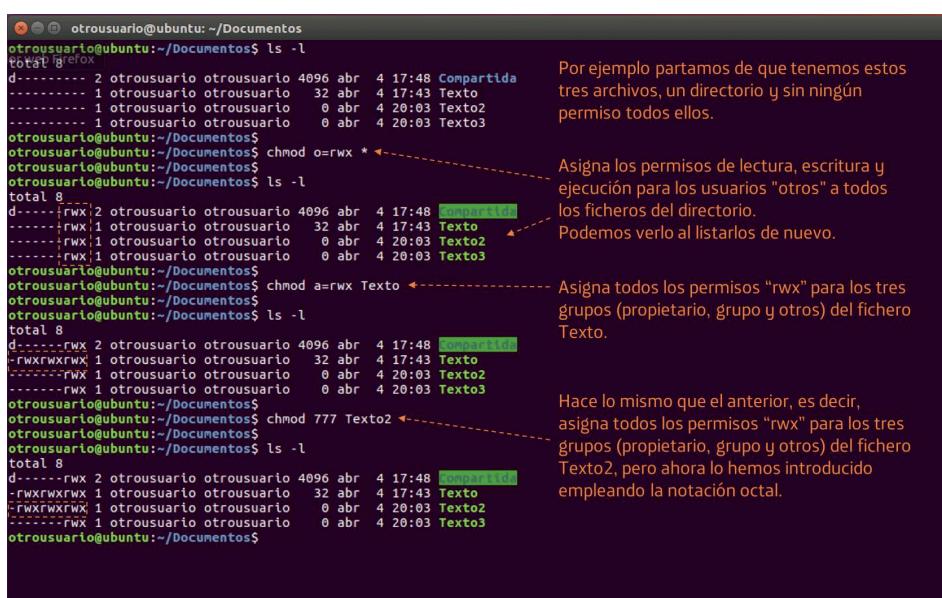
```
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwxr-x 2 otousuario otousuario 4096 abr 4 17:48 Compartida
-rw-r--r-- 1 root      root        0 abr 4 17:48 Otrtexto
-rw-rw-r-- 1 otousuario otousuario 32 abr 4 17:43 Texto
-rw-rw-r-- 1 otousuario otousuario 0 abr 4 20:03 Texto2
-rw-rw-r-- 1 otousuario otousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod 000 Texto
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwxr-x 2 otousuario otousuario 4096 abr 4 17:48 Compartida
-rw-r--r-- 1 root      root        0 abr 4 17:48 Otrtexto
----- 1 otousuario otousuario 32 abr 4 17:43 Texto
----- 1 otousuario otousuario 0 abr 4 20:03 Texto2
----- 1 otousuario otousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod 000 Texto2
otrousuario@ubuntu:~/Documentos$ chmod 000 Texto3
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwxr-x 2 otousuario otousuario 4096 abr 4 17:48 Compartida
-rw-r--r-- 1 root      root        0 abr 4 17:48 Otrtexto
----- 1 otousuario otousuario 32 abr 4 17:43 Texto
----- 1 otousuario otousuario 0 abr 4 20:03 Texto2
----- 1 otousuario otousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod 000 Compartida
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
----- 2 otousuario otousuario 4096 abr 4 17:48 Compartida
-rw-r--r-- 1 root      root        0 abr 4 17:48 Otrtexto
----- 1 otousuario otousuario 32 abr 4 17:43 Texto
----- 1 otousuario otousuario 0 abr 4 20:03 Texto2
----- 1 otousuario otousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod 000 Otrtexto
otrousuario@ubuntu:~/Documentos$ chmod: cambiando los permisos de 'Otrtexto': Operación no permitida
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ 
```

Por ejemplo tenemos estos archivos y un directorio en nuestra carpeta de "Documentos".

Vamos a quitarle todos los permisos a un fichero. Basta con introducir este comando y vemos luego que ya no tiene ningún permiso.

Se los vamos a quitar al resto de los ficheros y al directorio también, con el mismo comando.

Fíjate que cuando se los intentamos quitar a "Otrtexto" no nos deja. ¿por qué? Pues porque no somos su dueño (es "root").



```
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
d----- 2 otousuario otousuario 4096 abr 4 17:48 Compartida
----- 1 otousuario otousuario 32 abr 4 17:43 Texto
----- 1 otousuario otousuario 0 abr 4 20:03 Texto2
----- 1 otousuario otousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod o=rwx *
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
d----rwx 2 otousuario otousuario 4096 abr 4 17:48 Compartida
-----rwx 1 otousuario otousuario 32 abr 4 17:43 Texto
-----rwx 1 otousuario otousuario 0 abr 4 20:03 Texto2
-----rwx 1 otousuario otousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod a=rwx Texto
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
d-----rwx 2 otousuario otousuario 4096 abr 4 17:48 Compartida
-----rwx 1 otousuario otousuario 32 abr 4 17:43 Texto
-----rwx 1 otousuario otousuario 0 abr 4 20:03 Texto2
-----rwx 1 otousuario otousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod a=rwx Texto2
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
d-----rwx 2 otousuario otousuario 4096 abr 4 17:48 Compartida
-----rwx 1 otousuario otousuario 32 abr 4 17:43 Texto
-----rwx 1 otousuario otousuario 0 abr 4 20:03 Texto2
-----rwx 1 otousuario otousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ 
```

Por ejemplo partamos de que tenemos estos tres archivos, un directorio y sin ningún permiso todos ellos.

Asigna los permisos de lectura, escritura y ejecución para los usuarios "otros" a todos los ficheros del directorio. Podemos verlo al listarlos de nuevo.

Asigna todos los permisos "rwx" para los tres grupos (propietario, grupo y otros) del fichero Texto.

Hace lo mismo que el anterior, es decir, asigna todos los permisos "rwx" para los tres grupos (propietario, grupo y otros) del fichero Texto2, pero ahora lo hemos introducido empleando la notación octal.

```
otrousuario@ubuntu:~/Documentos
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
d----- 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
----- 1 otrousuario otrousuario 32 abr 4 17:43 Texto
----- 1 otrousuario otrousuario 0 abr 4 20:03 Texto2
----- 1 otrousuario otrousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod u=rwx,g=rw,o= Texto*
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
d----- 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwxr----- 1 otrousuario otrousuario 32 abr 4 17:43 Texto
-rwxr----- 1 otrousuario otrousuario 0 abr 4 20:03 Texto2
-rwxr----- 1 otrousuario otrousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod 000 *
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
d----- 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
----- 1 otrousuario otrousuario 32 abr 4 17:43 Texto
----- 1 otrousuario otrousuario 0 abr 4 20:03 Texto2
----- 1 otrousuario otrousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod 760 Texto* ←
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
d----- 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwxrw---- 1 otrousuario otrousuario 32 abr 4 17:43 Texto
-rwxrw---- 1 otrousuario otrousuario 0 abr 4 20:03 Texto2
-rwxrw---- 1 otrousuario otrousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$
```

Para que sea más claro, volvemos a partir de una situación con todos los objetos sin permisos.

Otorga todos los permisos al propietario del fichero, da permisos de lectura y escritura a los del grupo del dueño, y ningún permiso a los otros usuarios del sistema.  
Además se ejecuta sobre todos los ficheros cuyo nombre empieza por "Texto" (es decir sobre los tres que tenemos).

Este comando hace lo mismo que el anterior chmod pero lo hemos escrito empleando notación octal.

```
otrousuario@ubuntu:~/Documentos
otrousuario@ubuntu:~/Documentos$ chmod 777 *
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwxrwx 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwxrwxrwx 1 otrousuario otrousuario 32 abr 4 17:43 Texto
-rwxrwxrwx 1 otrousuario otrousuario 0 abr 4 20:03 Texto2
-rwxrwxrwx 1 otrousuario otrousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod o-rwx Texto* ←
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwxrwx 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwxrwx--- 1 otrousuario otrousuario 32 abr 4 17:43 Texto
-rwxrwx--- 1 otrousuario otrousuario 0 abr 4 20:03 Texto2
-rwxrwx--- 1 otrousuario otrousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod u-x,g-wx,o-rwx Texto* ←
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwxrwx 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rw-r-- 1 otrousuario otrousuario 32 abr 4 17:43 Texto
-rwxrwx--- 1 otrousuario otrousuario 0 abr 4 20:03 Texto2
-rwxrwx--- 1 otrousuario otrousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ chmod u-x,g-wx,o-rwx Compartida ←
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxr-xr-x 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rw-r----- 1 otrousuario otrousuario 32 abr 4 17:43 Texto
-rwxrwx--- 1 otrousuario otrousuario 0 abr 4 20:03 Texto2
-rwxrwx--- 1 otrousuario otrousuario 0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$
```

Ahora vamos a quitar permisos, supongamos que todos los objetos tienen todos los permisos...  
(se los hemos dado con este comando).

Retiramos los permisos rwx (lectura, escritura y ejecución) a los usuarios "otros" (que no son de nuestro grupo) sobre los tres archivos Texto\*.

Actuamos solo sobre el fichero "Texto" y ...  
Retiramos el permiso de ejecución al propietario.  
Retiramos permiso de escritura y ejecución a los miembros de su grupo.  
Retiramos todos los permisos al resto (otros).

Al directorio "Compartida" le damos todos los permisos al dueño, permiso de lectura y ejecución para los miembros del grupo, y solamente permiso de ejecución para los demás usuarios.

**Te recomendamos que practiques bien con este comando sobre tu máquina virtual Ubuntu.**

# Ejemplos de permisos especiales en Linux

Como el concepto de los permisos especiales de Linux es más complejo que los permisos básicos que hemos manejado hasta ahora, vamos a verlos a través de unos ejemplos.

Debes esforzarte en entenderlos bien porque es un elemento muy importante para trabajar en este S.O.

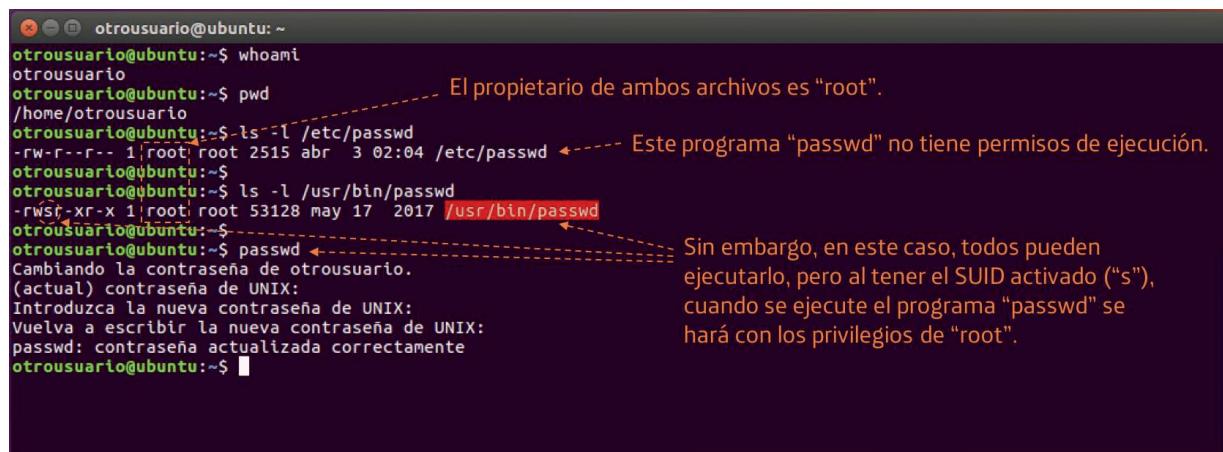
## ¿Cómo funciona el tema de los permisos especiales SUID y SGID?

Veamos el proceso a seguir, por ejemplo, si un **proceso** de un usuario (**UID1** y **GID1**) quiere acceder a un fichero/directorio que pertenece a otro propietario (con **UIDp** y **GIDp**):

1. ¿**UID1=UIDp**? (¿el que accede es el propietario?): se aplican los permisos del propietario, y si no...
2. ¿**GID1=GIDp**? (¿es del mismo grupo que el propietario?): se aplican los permisos del grupo **GIDp**, y si no...
3. Se aplican los permisos de “otros” (del resto de usuarios). Hasta aquí todo normal, pero además...
4. Si el fichero tiene permiso de ejecución, se trata de un ejecutable binario (no un *script*) y... ¿tiene SUID activado?: se ejecuta con los privilegios del propietario (**UIDp**).
5. ¿Y si tiene activado SGID?: entonces si es un ejecutable, el programa se ejecuta con los privilegios de su grupo (**GIDp**) con un comportamiento similar al anterior. Si en cambio es un directorio el que tiene el SGID activado, entonces lo que pasará es que cualquier otro directorio o fichero que un usuario cree dentro de él, pertenecerá al mismo grupo que el directorio padre (en el que nos encontramos y tiene el SGID activado).

## Un ejemplo de uso de SUID

El comando “passwd” sirve para cambiar la clave de usuario y todos los usuarios del sistema pueden ejecutarlo, pero cambiar la clave es un privilegio de un administrador, por lo que aunque un usuario tuviese permiso de lanzar su ejecución, dentro del S.O. el proceso arrancaría con la ID del usuario “normal” y puede ser que el programa intentase ejecutar acciones para las que no tuviese autorización. Para permitir usarlo, en el sistema tenemos también un programa “passwd” (en el directorio “/usr/bin/”) que puede ser ejecutado por todos los usuarios y que tiene “SUID” activado. Veámoslo con un ejemplo:



A terminal window showing a user named 'otrousuario' at the prompt. The user runs several commands to demonstrate SUID:

```
otrousuario@ubuntu:~$ whoami
otrousuario
otrousuario@ubuntu:~$ pwd
/home/otrousuario
otrousuario@ubuntu:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2515 abr  3 02:04 /etc/passwd
otrousuario@ubuntu:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 53128 may 17  2017 /usr/bin/passwd
otrousuario@ubuntu:~$ ./passwd
otrousuario@ubuntu:~$ passwd
Cambiando la contraseña de otrousuario.
(actual) contraseña de UNIX:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
otrousuario@ubuntu:~$
```

Annotations explain the SUID mechanism:

- Annotation 1: "El propietario de ambos archivos es "root".
- Annotation 2: "Este programa "passwd" no tiene permisos de ejecución."
- Annotation 3: "Sin embargo, en este caso, todos pueden ejecutarlo, pero al tener el SUID activado ("s"), cuando se ejecute el programa "passwd" se hará con los privilegios de "root".

- Para modificar el **SUID** se utiliza “chmod u+s fichero” o bien “chmod 4xxx fichero” (en octal), y para quitarlo “chmod u-s fichero”.
- Para modificar el **SGID** se utiliza “chmod g+s fichero” o bien “chmod 2xxx fichero” (en octal), y para quitarlo “chmod g-s fichero”.

## Un ejemplo de uso de SGID

Este puede ser un poco más difícil de comprender, por lo que te lo iremos describiendo paso a paso en una secuencia de comandos:

```

otrousuario@ubuntu:~/Documentos
otrousuario@ubuntu:~$ pwd
/home/otrousuario
otrousuario@ubuntu:~$ cd Documentos
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwxr-x 2 ourosario ourosario 4096 abr 4 17:44 Compartida
-rw-rw-r-- 1 ourosario ourosario   32 abr 4 17:43 Texto
otrousuario@ubuntu:~/Documentos$ chmod g+s Compartida
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwsr-x 2 ourosario ourosario 4096 abr 4 17:44 Compartida
-rw-rw-r-- 1 ourosario ourosario   32 abr 4 17:43 Texto
otrousuario@ubuntu:~/Documentos$ su
Contraseña:
root@ubuntu:/home/otrousuario/Documentos# cd Compartida
root@ubuntu:/home/otrousuario/Documentos/Compartida#
root@ubuntu:/home/otrousuario/Documentos/Compartida# touch Nuevodoc
root@ubuntu:/home/otrousuario/Documentos/Compartida# ls -l
total 0
-rw-r--r-- 1 root ourosario 0 abr 4 17:48 Nuevodoc
root@ubuntu:/home/otrousuario/Documentos/Compartida# cd ..
root@ubuntu:/home/otrousuario/Documentos# touch Otrotexto
root@ubuntu:/home/otrousuario/Documentos# ls -l
total 8
drwxrwsr-x 2 ourosario ourosario 4096 abr 4 17:48 Compartida
-rw-r--r-- 1 root root 0 abr 4 17:48 Otrotexto
-rw-rw-r-- 1 ourosario ourosario   32 abr 4 17:43 Texto
otrousuario@ubuntu:~/Documentos$ exit
exit
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxrwsr-x 2 ourosario ourosario 4096 abr 4 17:48 Compartida
-rw-r--r-- 1 root root 0 abr 4 17:48 Otrotexto
-rw-rw-r-- 1 ourosario ourosario   32 abr 4 17:43 Texto
otrousuario@ubuntu:~/Documentos$ ls -l ./Compartida
total 0
-rw-r--r-- 1 root ourosario 0 abr 4 17:48 Nuevodoc
otrousuario@ubuntu:~/Documentos$ 

```

Somos el usuario "otrousuario", y dentro de nuestro directorio personal tenemos el directorio "Documentos", dentro del cual hemos creado una carpeta (subdirectorio) llamada "Compartida" y un archivo ("Texto") que no usaremos (solo para que hubiese algo más).

Asignamos y activamos el SGID a la carpeta "Compartida" y vemos que efectivamente aparece con ese permiso especial.

El propietario de esta carpeta es "otrousuario" y pertenece a su grupo.

Ahora vamos a actuar como "root" (es decir, como otro usuario) y crearemos un nuevo archivo ("Nuevodoc") dentro de la carpeta "Compartida". *Nota: Como "truco" usamos el comando "touch" que creará un fichero vacío.*

Fíjate que lo ha creado "root" (propietario) pero le ha asignado el mismo grupo al que pertenecía la carpeta "Compartida", es decir, "otrousuario".

En cambio, si creamos otro fichero ("Otrotexto") como usuario "root", pero fuera de la carpeta que tiene el SGID activado, por ejemplo en "Documentos", lo creará con root como propietario y grupo.

Evidentemente si volvemos a visualizar el fichero "Nuevodoc", pero siendo "otrousuario" nos dice que aunque es de nuestro grupo lo ha creado "root" (es su propietario).



**Nota:** El comando "touch" que hemos utilizado sirve para actualizar los registros de fecha y hora, con la fecha y hora actual en un archivo que se pasa como argumento, pero si se deja vacío lo crea. Es un pequeño "truco" que hemos utilizado, pero no debes centrarte ahora en este comando.

## ¿Y cómo se usa el "sticky" bit?

Se suele asignar a los directorios sobre los que todo el mundo puede escribir y crear archivos, para que solamente el propietario del archivo (y root por supuesto) pueda borrarlo. De esta forma evitamos, por ejemplo, que "Pepe" cree un archivo en "/tmp" y luego "Ramón" lo borre por error. El comando para asignarlo es similar: "chmod o+t directorio".

```
otrousuario@ubuntu:~/tmp
otrousuario@ubuntu:~$ pwd
/home/otrousuario
otrousuario@ubuntu:~$ ls -l / | grep "tmp"
drwxrwxrwt 12 root root 4096 abr 4 18:29 tmp
otrousuario@ubuntu:~$ 
otrousuario@ubuntu:~$ ls -l /tmp | grep "textodepepe"
-rw-rw-r-- 1 pepe      pepe        0 abr 4 18:29 textodepepe
otrousuario@ubuntu:~$ cd /tmp
otrousuario@ubuntu:/tmp$ touch textomio
otrousuario@ubuntu:/tmp$ ls -l /tmp | grep "texto*"
-rw-rw-r-- 1 pepe      pepe        0 abr 4 18:29 textodepepe
-rw-rw-r-- 1 otrousuario otrousuario 0 abr 4 18:31 textomio
otrousuario@ubuntu:/tmp$ 
otrousuario@ubuntu:/tmp$ rm textomio <----- Vemos que nos deja borrar el nuestro pero en cambio no nos deja borrar el de Pepe.
otrousuario@ubuntu:/tmp$ ls -l /tmp | grep "texto*"
-rw-rw-r-- 1 pepe      pepe        0 abr 4 18:29 textodepepe
otrousuario@ubuntu:/tmp$ rm textodepepe
rm: ¿borrar el fichero regular vacío 'textodepepe' protegido contra escritura? (s/n) s
rm: no se puede borrar 'textodepepe': Operación no permitida
otrousuario@ubuntu:/tmp$
```

Visualizamos el directorio "/tmp" que tiene el Sticky bit activado.

En "/tmp" tenemos un fichero que ha creado Pepe, nosotros también crearemos otro ("textomio") y luego intentamos borrar los dos.

Vemos que nos deja borrar el nuestro pero en cambio no nos deja borrar el de Pepe.

## ¿Y cómo usar "chmod" con los permisos especiales?

Podemos usarlo de igual forma que con los permisos básicos, tanto en notación simbólica como en notación octal (pero en este caso usaremos cuatro dígitos). Te lo mostramos con unos ejemplos:

```
otrousuario@ubuntu:~/Documentos
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxr-x--- 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwxrwxr-- 1 otrousuario otrousuario   32 abr 4 17:43 Texto
-rwxrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto2
-rwxrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ chmod g+s Compartida <----- Veamos ahora como usar "chmod" para asignar los permisos especiales...
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxr-s---x 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwxrwxr-- 1 otrousuario otrousuario   32 abr 4 17:43 Texto
-rwxrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto2
-rwxrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ chmod u+s Texto <----- Este comando activa el SGID para el directorio "Compartida".
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxr-s---x 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwsrwxr-- 1 otrousuario otrousuario   32 abr 4 17:43 Texto
-rwsrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto2
-rwsrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ chmod 4774 Texto2 <----- Este comando activa el SUID para el fichero "Texto" (que aunque se llama así. está vacío y no es de texto, puede ser ejecutable).
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxr-s---x 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwsrwxr-- 1 otrousuario otrousuario   32 abr 4 17:43 Texto
-rwsrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto2
-rwsrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ chmod 774 Texto2 <----- Este comando activa el SUID para el fichero "Texto2", igual que el anterior pero usando notación octal con cuatro dígitos (el primer dígito es el que activa el SUID).
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$ ls -l
total 8
drwxr-s---x 2 otrousuario otrousuario 4096 abr 4 17:48 Compartida
-rwsrwxr-- 1 otrousuario otrousuario   32 abr 4 17:43 Texto
-rwsrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto2
-rwsrwxr-- 1 otrousuario otrousuario    0 abr 4 20:03 Texto3
otrousuario@ubuntu:~/Documentos$ 
otrousuario@ubuntu:~/Documentos$
```

Este comando activa el SGID para el directorio "Compartida".

Este comando activa el SUID para el fichero "Texto" (que aunque se llama así. está vacío y no es de texto, puede ser ejecutable).

Este comando activa el SUID para el fichero "Texto2", igual que el anterior pero usando notación octal con cuatro dígitos (el primer dígito es el que activa el SUID).

Con este comando, usando notación octal, le retiramos el SUID al usar solamente tres dígitos, pero conservamos el resto de los permisos del fichero.

## Resumen

---

Has finalizado esta unidad. Repasemos los puntos más importantes que hemos tratado.

- Conocer la configuración de perfiles locales de usuario, así como los diferentes tipos de perfiles y cómo podemos modificar el perfil de un usuario.
- Aprender que los directorios y archivos relacionados con la información de los perfiles de usuario son diferentes de un sistema operativo a otro, y que también tienen diferentes ubicaciones en Windows y Linux (Ubuntu).
- Es importante saber que en un sistema multiusuario cada uno de ellos tendrá una serie de derechos y privilegios que le permiten realizar ciertas acciones, así como una serie de permisos.



**PRO**EDUCA