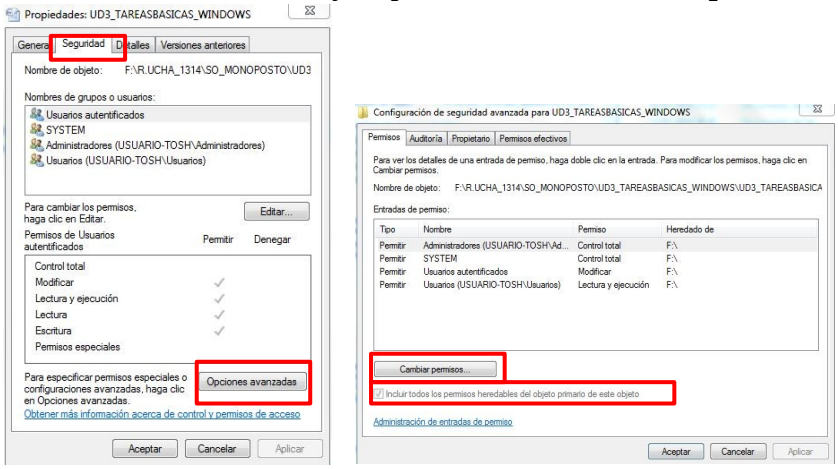
Asignar permisos en entorno gráfico

Establecer y cambiar permisos en Windows 10

En los sistemas de archivos suele compartirse información con otros usuarios, por ello es necesario controlar el acceso a los archivos. Un permiso indica qué cosas puede hacer un usuario o grupo de usuarios en cualquier archivo. Los permisos se pueden **agregar o denegar a un usuario o grupo se usuarios**. Los seis permisos estándar para el sistema de ficheros NTFS son:

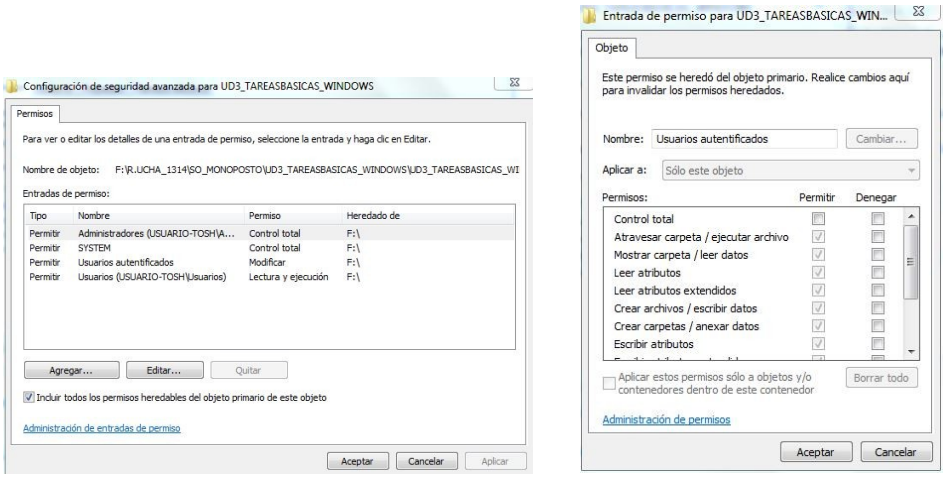
- **Control total:** los archivos y directorios se pueden leer, escribir, cambiar y eliminar. En el caso de los archivos además se pueden cambiar permisos (modificar + cambiar permisos)
- **Modificar:** se pueden leer, escribir, cambiar y eliminar archivos y directorios.
- **Lectura y ejecución:** se puede acceder a los ficheros y ejecutarlos. En el caso de las carpetas, se puede ver el contenido de una carpeta y ejecutar.
- **Lectura:** Se puede ver el contenido de los archivos pero no se pueden ejecutar. En el caso de las carpetas se puede acceder al contenido.
- **Escritura:** los archivos se pueden modificar pero no eliminar. En los directorios se pueden añadir ficheros y subcarpetas.
- **Mostrar el contenido:** para los directorios se puede mostrar el contenido.

De forma **predeterminada**, los objetos de un contenedor **heredan los permisos de ese contenedor** al ser creados. Los permisos heredados se pueden cambiar desde la configuración avanzada de permisos.
Para acceder a los permisos de un archivo se pulsa con el botón derecho sobre él, se selecciona la opción de **Propiedades**, la pestaña de **seguridad** y se pulsa el botón **Opciones avanzadas**. En la pestaña **Permisos** se selecciona un usuario y se pulsa el botón **Cambiar permisos**.



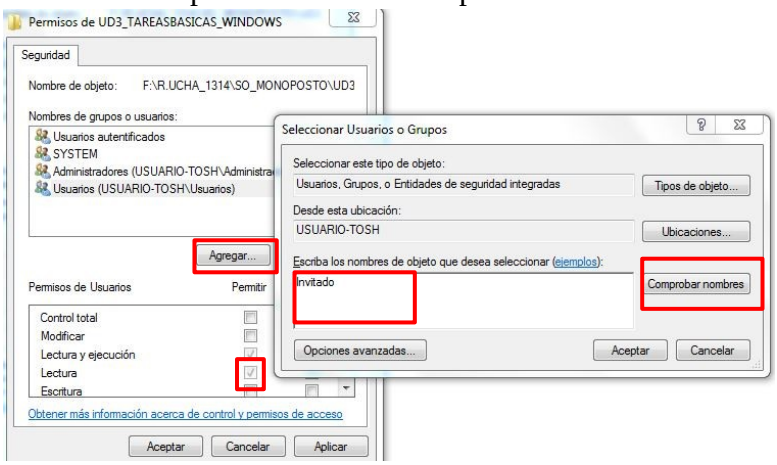
Permisos de un archivo

Los permisos heredados se pueden cambiar desactivando la opción "**Incluir todos los permisos heredables del objeto primario de este objeto**".
En la nueva pantalla se selecciona un usuario y se pulsa **Editar**, acto seguido aparecerá una ventana con los distintos permisos especiales que se pueden permitir o denegar.



Permitir o denegar permisos

Añadir un usuario y cambiarle los permisos también se puede hacer desde la ventana principal.



Asignar permiso de lectura al usuario Invitado

Establecer y cambiar permisos en Ubuntu

En Linux los tipos de usuarios son:

- **Propietario (owner):** creador del archivo
- **Grupo (group):** conjunto de usuarios
- **Resto de usuarios (others):** usuarios que no pertenecen a un grupo ni son propietarios.

Los permisos en el sistema de ficheros ext4 son:

- **Lectura (r, read):** ver e imprimir archivos; se pueden ver todos los elementos del directorio.
- **Escritura (w, write):** cambiar o eliminar archivos o directorios.
- **Ejecución (x, execute):** el fichero puede ser ejecutado.

Para cambiar permisos a un archivo hay que acceder al **menú contextual** y pulsar en la opción de **propiedades** y a continuación en la pestaña de **permisos**. Pulsando en "**cambiar permisos a los archivos contenidos**" se abre una nueva ventana.
En las opciones de archivos aparecen permisos diferentes a los de carpetas. En la opción del propietario, tanto en archivos como en carpetas, no aparece la opción de "ninguno".

Permisos en modo gráfico. Permisos de archivos y permisos de carpetas

Establecer y cambiar permisos por línea de comandos

Cada archivo en Linux queda identificado por **diez caracteres** (**máscara**).

- El **primer carácter** empezando por la izquierda indica el **tipo de archivo**. Los posibles tipos son: **Normal (-), directorio (d), enlace simbólico (l), entrada y salida (c,b,s,p).**

- Los **nueve caracteres** siguientes organizados en **conjuntos de tres** indican los **permisos** para cada categoría de usuarios. Las **categorías de usuarios** (empezando por la izquierda) son propietario, grupo y resto de usuarios.

```
uadmin@uclient01:~/Documentos$ ls -l
total 4
-rw-rw-r-- 1 uadmin uadmin    0 may 15 16:11 apuntes
-rw-rw-r-- 1 uadmin uadmin    0 may 15 16:11 fechas
drwxrwxr-x 2 uadmin uadmin 4096 may 15 16:10 fotos
uadmin@uclient01:~/Documentos$
```

Archivos normales y directorio

Para **asignar y cambiar permisos** se utiliza el comando: **chmod [opciones] fichero**
Existen dos formas de asignar permisos, en octal o utilizando letras.

- En **octal** se utilizan 3 dígitos para especificar los permisos. El primero para especificar los permisos del propietario, otro para los permisos del grupo y el último para los permisos de otros usuarios. El valor de cada uno de esos tres dígitos se calcula teniendo en cuenta el orden de permisos (rwx). Si se asigna permiso se utilizará un 1, si no se asigna se utilizará un 0. A continuación se hará la conversión de binario a octal.

  Ejemplo: 111(usuario) = 7, 110(grupo) = 6, 101(otros) = 5, chmod 765 ejemplo.txt
- Utilizando **letras** se especifica quien, la operación y los permisos

  - quien: u (usuarios), g (grupo), o (otros), a (todos)
  - operación: + (añadir) y - (eliminar), = (asignar).
  - permisos: r (lectura), w (escritura), x (ejecución)

Ejemplo: rwx al usuario, rw al grupo y r a otros. chmod u+rwx,g+rw,o+r ejemplo.txt

```
patricia@cliubupatri:~$ chmod 765 one.txt
patricia@cliubupatri:~$ chmod u+rwx,g+rw,o+r one.txt
patricia@cliubupatri:~$
```

Asignación de permisos con chmod

Cambia los permisos a un archivo de modo que el propietario tenga todos los permisos, el grupo lectura y ejecución y otros solo lectura

```
chmod 754 archivo.txt
```

```
chmod u=rwx,g=rx,o=r archivo.txt
```

Añade permisos de escritura de grupo al fichero ej.txt

```
chmod g+w ej.txt
```

Quita los permisos de ejecución a los otros usuarios, al fichero ej2.txt

```
chmod o-x ej2.txt
```

Permisos especiales

Existen unos bits que ofrecen la posibilidad de tener unas medidas de protección adicional ajustando permisos especiales sobre ficheros y directorios:

- **sticky bit** se indica con el carácter **t**. Se utilizaba antiguamente en sistemas UNIX para conseguir que el sistema operativo mantuviera en memoria los programas que tenían el sticky bit activado. Hoy en día, el sticky bit en un directorio significa que **tan solo los respectivos dueños de los archivos que haya en el directorio y el superusuario pueden borrarlos**. El mayor beneficio de este permiso se obtiene colocándole en los directorios públicos (aquellos en los que cualquier usuario puede depositar archivos), con lo cual **se evita que usuarios distintos al propietario alteren el archivo colocado en un directorio**.

- **SUID(Set-user Identification)** se representa por **s** y normalmente se activa en ejecutables. Cuando a un ejecutable binario se le asigna el atributo setuid, **usuarios normales del sistema pueden ejecutar ese archivo y obtener privilegios del usuario que posee dicho archivo** (generalmente, root) en el proceso creado. Por ejemplo el bit suid se utiliza en el fichero **/usr/bin/passwd** para que todo el mundo pueda cambiar su contraseña de forma controlada. Pudiendo ejecutar este programa se consigue que un usuario pueda escribir en el fichero de claves (/etc/shadow, no confundirlo con /etc/passwd) pero sin tener que dar permisos de escritura al fichero, lo cual sería un gran agujero de seguridad.

- **SGID (Set-group identification)** se representa por **s** y es **lo mismo que en el SUID, pero a nivel de grupo**. Es decir, todo archivo que tenga activo el SGID, al ser ejecutado, tendrá los privilegios del grupo al que pertenece. Es utilizado cuando se quiere configurar un **directorio colaborativo**: si se aplica este bit al directorio, cualquier archivo creado en dicho directorio, tendrá asignado el grupo al que pertenece el directorio.

Si en vez de una **s** o una **t**, aparece una S o T(mayúscula), significa que el **permiso no es efectivo**, porque le **falta el permiso de ejecución al usuario, al grupo o a los otros**. Es decir, para que sea efectivo el permiso debe tener permisos de ejecución.
Para asignar estos permisos se puede usar el modo simbólico o el modo octal.

- En modo octal:

  chmod 4xxx fichero para activar el modo **suid**
  chmod 2xxx fichero para activar el modo **sgid**
  chmod 1xxx fichero para activar el modo **sticky bit**

- En modo simbólico:

  chmod u+s fichero para activar el modo **suid**
  chmod g+s fichero para activar el modo **sgid**

chmod o+t fichero para activar el **sticky bit**

Ejemplo con permisos 666. Establecer los permisos de chmod como:

Propietario	Suid: rwsr--r--(4)	Permisos en octal 4744
Grupo	Sgid: rwxrwsr--(2)	Permisos en octal 2774
Otros	Sticky bit: rwxr-xrwt(1)	Permisos en octal 1757