

**MP0483.
Sistemas informáticos**

UF6. Gestión de recursos en una red

6.1. Permisos y derechos

Índice

≡	Objetivos	3
≡	Derechos de los usuarios	4
≡	Ejemplos de derechos de usuarios	6
≡	Diferencia entre permisos y derechos	9
≡	Permisos locales	11
≡	Permisos de red	12
≡	Herencia	14
≡	Ejemplo de herencia de permisos en Windows	15
≡	Conflictos entre permisos	21
≡	Permisos efectivos	23
≡	Delegación de permisos	25
≡	Resumen	28

Objetivos

Ya sabemos que la seguridad del sistema, junto con el rendimiento y la disponibilidad de los servicios, es quizás el elemento más importante del que debe preocuparse un administrador.

En los sistemas en red la gestión de la seguridad va más allá de la simple configuración local de accesos y permisos de usuario. Es necesario definir una estrategia de configuración de seguridad a nivel de todos los equipos de la red, sean estos simples terminales, estaciones de trabajo, servidores o equipos de comunicaciones.

Por ello, nos parece conveniente que profundicemos un poco más en la diferencia entre permisos y derechos, entre permisos locales y en red, y qué hacer cuando surgen conflictos entre ellos.

En esta unidad perseguimos los siguientes objetivos:

- 1 Profundizar en la gestión y configuración de los perfiles de usuario, sus permisos y derechos de acceso al sistema.
- 2 Entender algunos conceptos importantes, como por ejemplo la herencia o la interacción entre permisos.
- 3 Repasar algunas actividades importantes a realizar por un administrador del sistema.

Derechos de los usuarios

En un sistema operativo en red los equipos se encuentran conectados y comparten recursos e información del sistema. Además, cada ordenador mantendrá su propio sistema operativo y su propio sistema de archivos local.

Los derechos de usuario son **tareas u operaciones que el usuario puede realizar** (tiene permiso y capacidad para ello) en el sistema, o en un dominio del equipo.

En general, en sistemas Windows existen dos tipos de derechos de usuario:

- **Derechos de inicio de sesión:** sirven para controlar quién puede iniciar sesión en el equipo y cómo lo hace. Un ejemplo puede ser el derecho a iniciar sesión de forma local o remota en el equipo.
- **Privilegios:** también se les llama “**permisos**” o “**derechos de acceso**”, se utilizan para gestionar el acceso a recursos del sistema y pueden definirse en conjunto o para objetos determinados (y entonces estos prevalecen sobre los del conjunto).



Los derechos de usuario los asigna el administrador, bien a usuarios individuales o a grupos, como parte de la configuración de seguridad del sistema.

Ejemplos de derechos de usuarios

Normalmente, el hecho de otorgar un derecho a un usuario lleva aparejado un cierto nivel de riesgo si ese usuario no actúa correctamente, o incluso aunque lo haga, por la propia exposición del equipo a eventos relacionados con su actuación y ese derecho.

Algunos ejemplos pueden ser los siguientes:

Derecho a acceder al equipo desde la red

Este derecho permite a un usuario conectarse al equipo desde la red. Será necesario tener este derecho si se quiere tener acceso a recursos compartidos en el equipo (impresoras o carpetas compartidas por ejemplo). Si se otorga este derecho al grupo “todos” y en el equipo hay carpetas compartidas, entonces cualquier usuario del sistema podrá verlas, lo cual puede representar un riesgo. En realidad el derecho a “acceder a este equipo desde la red” deben tenerlo solamente los usuarios que deban acceder a él como un servidor, como por ejemplo los del grupo “administradores”.

Ejecución en modo “kernel” o administrador

Se puede otorgar el derecho a “actuar como parte del sistema operativo”, pero hay que tener en cuenta que un usuario que pueda ejercerlo podrá a su vez modificar otros privilegios y actuar con un **alto riesgo** para el sistema. Este derecho debe limitarse al menor número de cuentas de usuario posibles, incluso no debería estar disponible para el grupo “administradores”, a no ser que fuese estrictamente necesario. Se puede solventar, cuando sea necesario, permitiendo el uso de una determinada cuenta (por ejemplo “system”) que tenga otorgado el permiso, sin asignar el permiso a otras cuentas.

Agregar estaciones de trabajo al dominio

Permite a un usuario agregar estaciones de trabajo a un determinado dominio, para lo cual el usuario normalmente debe formar parte del grupo de “controladores del dominio”, y no deben tener este derecho usuarios que no tengan funciones de administración.

Modificar las cuotas de memoria de los procesos

Este derecho permite al usuario que lo tiene poder ajustar la **cuota de memoria máxima** a utilizar por un proceso en ejecución en el sistema. Si un usuario con este derecho reduce excesivamente la cantidad de memoria disponible para un proceso, puede provocar fallos en la ejecución de aplicaciones o hacer que algunas de ellas funcionen con demasiada lentitud. Este derecho solamente deben tenerlo aquellos usuarios que realmente lo necesiten para realizar su trabajo, como por ejemplo los administradores de aplicaciones o bases de datos, o los administradores del dominio.

Iniciar sesión localmente en el equipo

Es el derecho que debe tener el usuario para **iniciar sesión local** interactiva en el equipo. Si el usuario no tiene este derecho, pero tiene el de **iniciar sesión remota**, entonces podría hacerlo aunque no fuese localmente, por lo que habrá que gestionar ambos derechos de forma coherente. Si hablamos de equipos de mucha importancia en el sistema y no de uso general (como pueden ser servidores o estaciones de administración), este derecho no debe estar disponible para todos los usuarios, sino para los del grupo de administración. También se pueden agregar cuentas específicas para ciertas labores, con derechos limitados, y permitir su uso por el grupo de usuarios que lo necesite.

Hacer copias de seguridad de ficheros y directorios

En general permite al usuario hacer una copia de seguridad del sistema, **por encima de los permisos de lectura/escritura de los archivos y directorios** (que son los que se tienen en cuenta normalmente), por lo que solamente será necesario cuando una aplicación intente hacer una copia de seguridad a través de las opciones de la interfaz de programación del propio sistema operativo (API), por ejemplo con comandos de *backup* disponibles en el propio sistema operativo.

Cambiar la hora y fecha del sistema

Como su nombre indica, permite al usuario ajustar la hora y fecha del reloj interno del equipo. En cambio, no suele ser necesario para cambiar la zona horaria u otros parámetros relacionados. Aunque puede no parecerlo, el hecho de poder cambiar la hora o fecha del equipo puede aparejar **riesgos importantes** y si no se hace correctamente puede provocar graves problemas. Alterar el reloj interno provocaría que las marcas de tiempo empleadas por las aplicaciones y los procesos de gestión no fuesen correctas y podría haber incoherencias entre distintos equipos en la red. Para solucionarlo, en muchos sistemas en red existe un servicio que sincroniza automáticamente la hora en todos los equipos.

Crear un archivo de paginación

Un **archivo de paginación** (oculto normalmente en el disco duro) se utiliza para facilitar el intercambio ("**swapping**") de páginas de memoria entre la RAM (memoria física) y la memoria virtual de los procesos presentes en el sistema.

Al otorgar este derecho permitimos al usuario crear un archivo de paginación y cambiar su tamaño. El riesgo asociado es que los usuarios que puedan cambiarlo lo hagan muy pequeño o moverlo a un volumen de almacenamiento que esté muy fragmentado. Ello podría reducir el rendimiento del sistema. Este derecho solamente debería estar otorgado al grupo de administradores del sistema.

En cada sistema, **el número de derechos otorgables a los usuarios dependerá de lo que nos ofrezca el propio sistema operativo**, y en general pueden ir ligados en mayor o menor medida a los derechos de administración del sistema. Así, podrán asignarse derechos, por ejemplo, para aumentar la prioridad de los procesos, cargar y actualizar controladores de dispositivo, forzar el apagado remoto de un equipo, bloquear la paginación en memoria, administrar registros y auditorías de seguridad, realizar tareas de mantenimiento, restaurar ficheros y directorios, etc.

Diferencia entre permisos y derechos

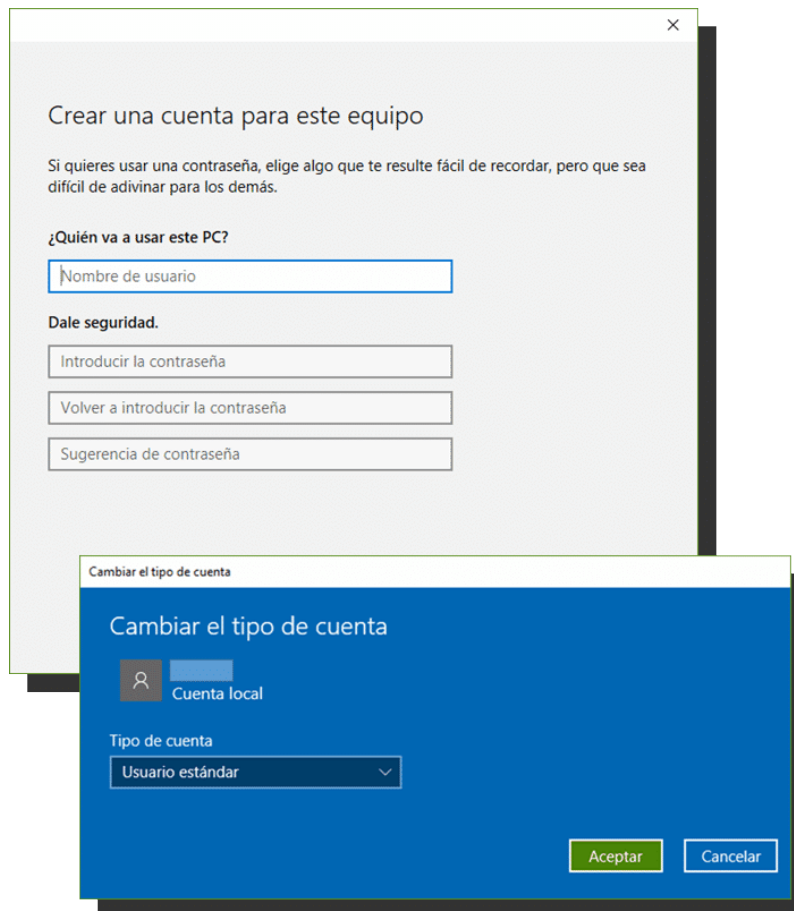
Un **derecho** autoriza al usuario a realizar ciertos procesos u operaciones. Por ejemplo, un usuario puede tener derecho a iniciar sesión, terminarla, realizar copias de seguridad, etc. Los **permisos**, en cambio, suelen dar acceso a recursos en el sistema o la red y están asociados a ese recurso.

Al otorgar un **permiso** lo que se concede, por lo tanto, es una **autorización para acceder a un recurso de red o un archivo**.

El **permiso** puede fijar o limitar la manera en la que debe realizarse el acceso al recurso y las acciones a realizar sobre él. Por ejemplo, un permiso puede autorizar a un usuario a acceder y leer un archivo, y sin embargo no darle licencia para modificarlo.

A menudo los **derechos** de una **cuenta local** incluyen el poder iniciar sesión y acceder localmente a los recursos (según los permisos que tengan asignados), cerrar la sesión y acceder a ese equipo desde la red.

Los derechos de una cuenta de usuario en un dominio permiten asimismo iniciar sesión en el dominio y acceder a los objetos dentro del dominio, también en función de los permisos que tenga asignados.



En entorno Windows los **permisos** están relacionados con **objetos**, mientras que los **derechos** se asignan a **cuentas** de usuario y de grupo y se aplican a través de "**directivas de grupo**" ("*Group Policy Object*" – GPO), como veremos más adelante.

Por ejemplo, un usuario puede tener derecho a realizar copias de seguridad del sistema (porque entre dentro de sus funciones) pero carecer de permisos de acceso sobre algunas carpetas del directorio.

Permisos locales

Los permisos locales marcan el nivel de acceso sobre los recursos locales del equipo y si pueden ser compartidos con otros usuarios.

En general, podrán **establecerse varios niveles de permisos** y debemos estudiar lo que nos ofrece cada sistema.

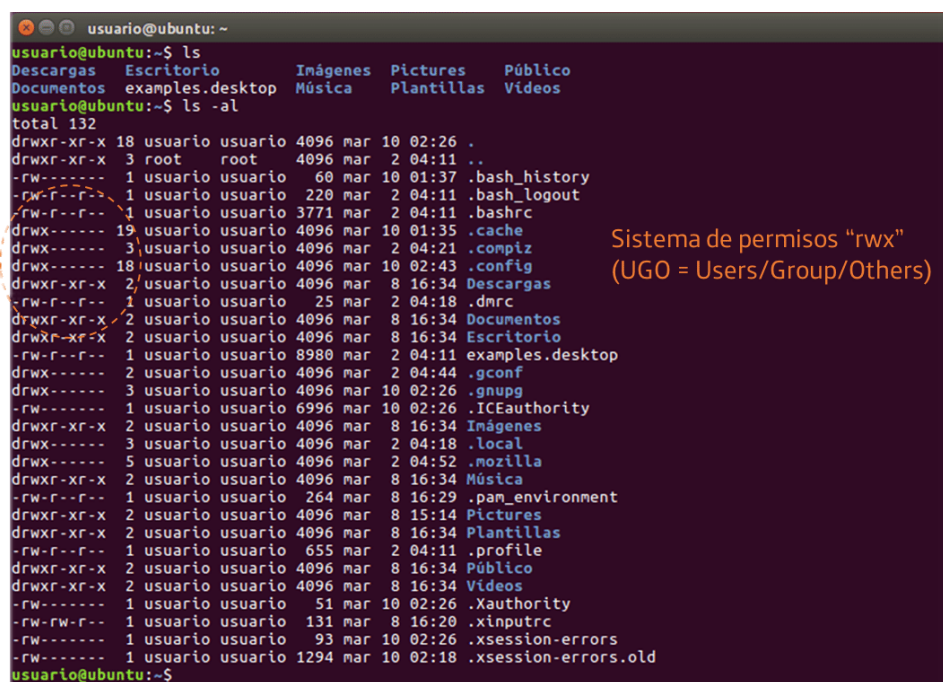
Por ejemplo, para el caso de **Windows** podrían ser:

- **Control total:** el usuario puede realizar cualquier acción, como visualizar, modificar archivos y carpetas, eliminarlos, crear nuevos, ejecutar programas, etc.
- **Modificar:** el usuario puede hacer cambios sobre los ficheros y carpetas existentes, pero no añadir o crear otros nuevos.
- **Lectura:** pueden visualizar el contenido de carpetas y archivos.
- **Lectura y ejecución:** además de visualizar el contenido, pueden ejecutar programas de la carpeta.
- **Escritura:** el usuario puede modificar los ficheros y carpetas y además puede crear otros nuevos.
- **Mostrar el contenido de la carpeta:** similar al de “lectura y ejecución”, pero con diferentes características de “herencia” (de lo que hablaremos después).

Permisos de red

A la hora de establecer los permisos sobre los diferentes objetos del sistema y la red (directorios, carpetas, equipos, etc.) la gestión puede variar bastante de un sistema operativo a otro.

Por ejemplo, en sistemas **Unix/Linux**, tanto a nivel local como en red, se utiliza el método **UGO**, ya conocido, basado en un esquema de permisos **rwX** (lectura, escritura, ejecución) asignables al usuario propietario del archivo, al grupo al que pertenece el propietario, o al resto de usuarios.



```
usuario@ubuntu: ~  
usuario@ubuntu:~$ ls  
Descargas Escritorio Inágenes Pictures Público  
Documentos examples.desktop Música Plantillas Videos  
usuario@ubuntu:~$ ls -al  
total 132  
drwxr-xr-x 18 usuario usuario 4096 mar 10 02:26 .  
drwxr-xr-x 3 root root 4096 mar 2 04:11 ..  
-rw----- 1 usuario usuario 60 mar 10 01:37 .bash_history  
-rw-r--r-- 1 usuario usuario 220 mar 2 04:11 .bash_logout  
-rw-r--r-- 1 usuario usuario 3771 mar 2 04:11 .bashrc  
drwx----- 19 usuario usuario 4096 mar 10 01:35 .cache  
drwx----- 3 usuario usuario 4096 mar 2 04:21 .complz  
drwx----- 18 usuario usuario 4096 mar 10 02:43 .config  
drwxr-xr-x 2 usuario usuario 4096 mar 8 16:34 Descargas  
-rw-r--r-- 1 usuario usuario 25 mar 2 04:18 .dmrc  
drwxr-xr-x 2 usuario usuario 4096 mar 8 16:34 Documentos  
drwxr-xr-x 2 usuario usuario 4096 mar 8 16:34 Escritorio  
-rw-r--r-- 1 usuario usuario 8980 mar 2 04:11 examples.desktop  
drwx----- 2 usuario usuario 4096 mar 2 04:44 .gconf  
drwx----- 3 usuario usuario 4096 mar 10 02:26 .gnupg  
-rw----- 1 usuario usuario 6996 mar 10 02:26 .ICEauthority  
drwxr-xr-x 2 usuario usuario 4096 mar 8 16:34 Inágenes  
drwx----- 3 usuario usuario 4096 mar 2 04:18 .local  
drwx----- 5 usuario usuario 4096 mar 2 04:52 .mozilla  
drwxr-xr-x 2 usuario usuario 4096 mar 8 16:34 Música  
-rw-r--r-- 1 usuario usuario 264 mar 8 16:29 .pam_environment  
drwxr-xr-x 2 usuario usuario 4096 mar 8 15:14 Pictures  
drwxr-xr-x 2 usuario usuario 4096 mar 8 16:34 Plantillas  
-rw-r--r-- 1 usuario usuario 655 mar 2 04:11 .profile  
drwxr-xr-x 2 usuario usuario 4096 mar 8 16:34 Público  
drwxr-xr-x 2 usuario usuario 4096 mar 8 16:34 Videos  
-rw----- 1 usuario usuario 51 mar 10 02:26 .Xauthority  
-rw-rw-r-- 1 usuario usuario 131 mar 8 16:20 .xinputrc  
-rw----- 1 usuario usuario 93 mar 10 02:26 .xsession-errors  
-rw----- 1 usuario usuario 1294 mar 10 02:18 .xsession-errors.old  
usuario@ubuntu:~$
```

Sistema de permisos "rwX"
(UGO = Users/Group/Others)

Método UGO de permisos en Unix.



Nota: Recuerda que en Linux “todo es un archivo”.

Si por el contrario estamos en un entorno **Windows** (moderno), deberemos trabajar con “**Directivas de grupo**” (“*Group Policy Object*” – GPO), que veremos más adelante, y es completamente distinto al anterior.

Como vemos, tanto la terminología como la forma de definir los permisos puede ser muy diferente de un sistema a otro, pero no debemos perder de vista el objetivo: organizar los recursos disponibles en la red y la actividad sobre ellos de los diferentes tipos de usuarios.

¿Qué son las “listas de control de acceso”?

En algunos sistemas, los permisos asociados a un objeto también pueden guardarse en unas “tablas” conocidas como “**ACL**” (“*Access Control Lists*”) o “Listas de control de acceso”. Estas ACL se utilizan para la gestión de los permisos asociados a un determinado recurso local o en red, e incluso pueden utilizarse, por ejemplo, para gestionar el tráfico a través de algunos elementos de la red (p. ej. *routers*).

Existen diversas formas de configurar una ACL: por usuario, por grupo, mediante una máscara de permisos, a través de diferentes comandos, etc. Su gestión es bastante compleja y no la abordaremos en este curso, pero queremos simplemente que conozcas su existencia y para qué sirven, por si en el futuro necesitas aprender sobre ellas.

Herencia

Hablamos de herencia cuando un elemento (objeto, fichero, carpeta, etc.) asume los atributos de otro de forma automática por ser jerárquicamente dependiente de él.

Por ejemplo, la **herencia de permisos** permite hacer la asignación una sola vez a un recurso, y que se apliquen a todos los que dependan de él y los hereden. La facultad de heredar los permisos puede desactivarse.

Siempre dependiendo del sistema operativo, a menudo la herencia se encuentra predeterminada para los objetos que dependan de otros en la jerarquía de la arquitectura lógica del sistema. En el caso de existir “herencia”, los derechos de acceso y permisos asignados a un “objeto primario” se extienden a los “objetos secundarios” que dependen de este. Hablamos entonces de “**permisos explícitos**” (los que se establecen sobre el objeto primario) y “**permisos heredados**” (los que se propagan a los objetos secundarios).

Por ejemplo, en el caso de Windows, de forma predeterminada, los objetos de un contenedor (o carpeta) heredan los permisos de ese contenedor cuando se crean esos objetos.

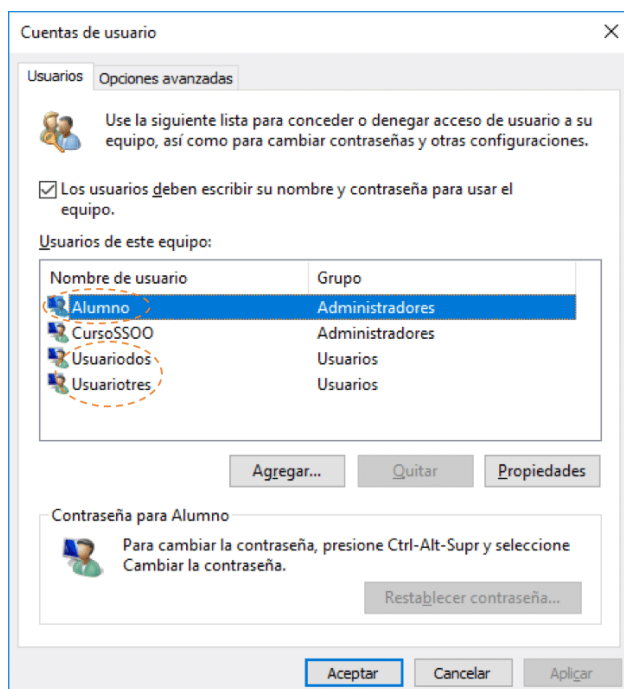


IMPORTANTE: en general, en el caso de **existir conflicto** entre ambos, **los permisos explícitos prevalecen sobre los permisos heredados**. Por ejemplo, si una carpeta hereda un atributo de “denegar el acceso”, pero lo tiene explícitamente permitido, se podrá acceder a ella.

Ejemplo de herencia de permisos en Windows

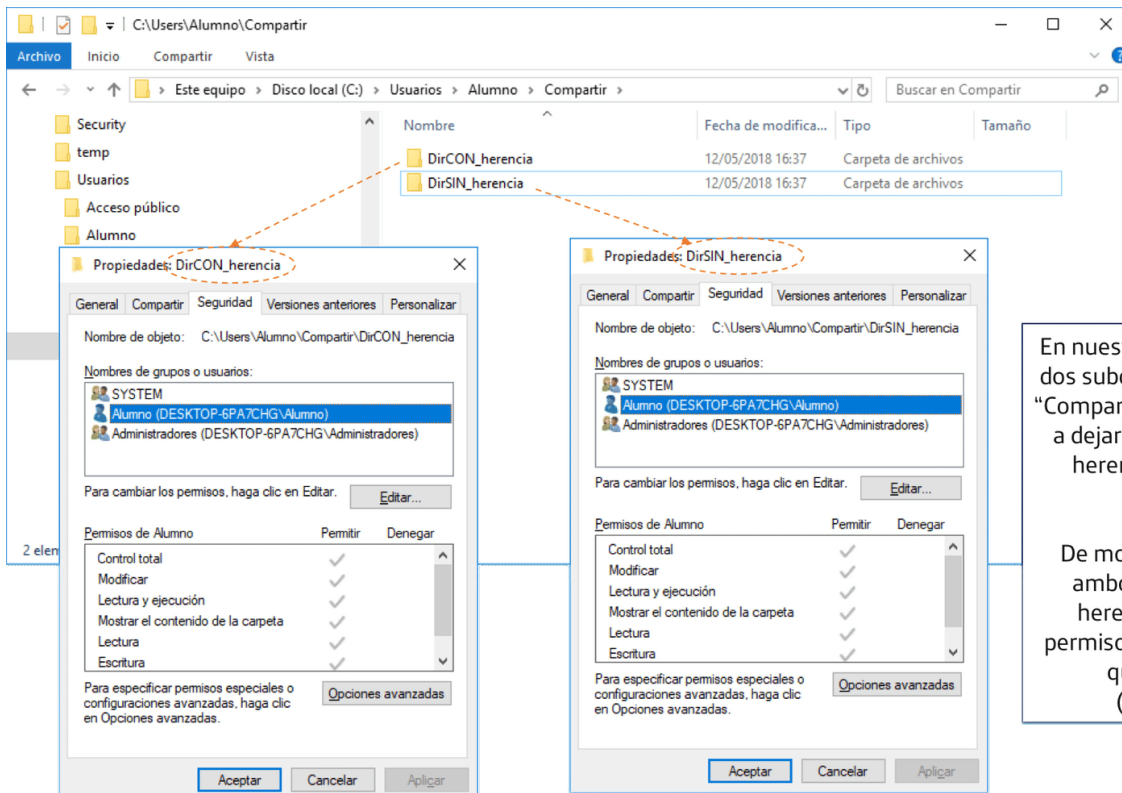
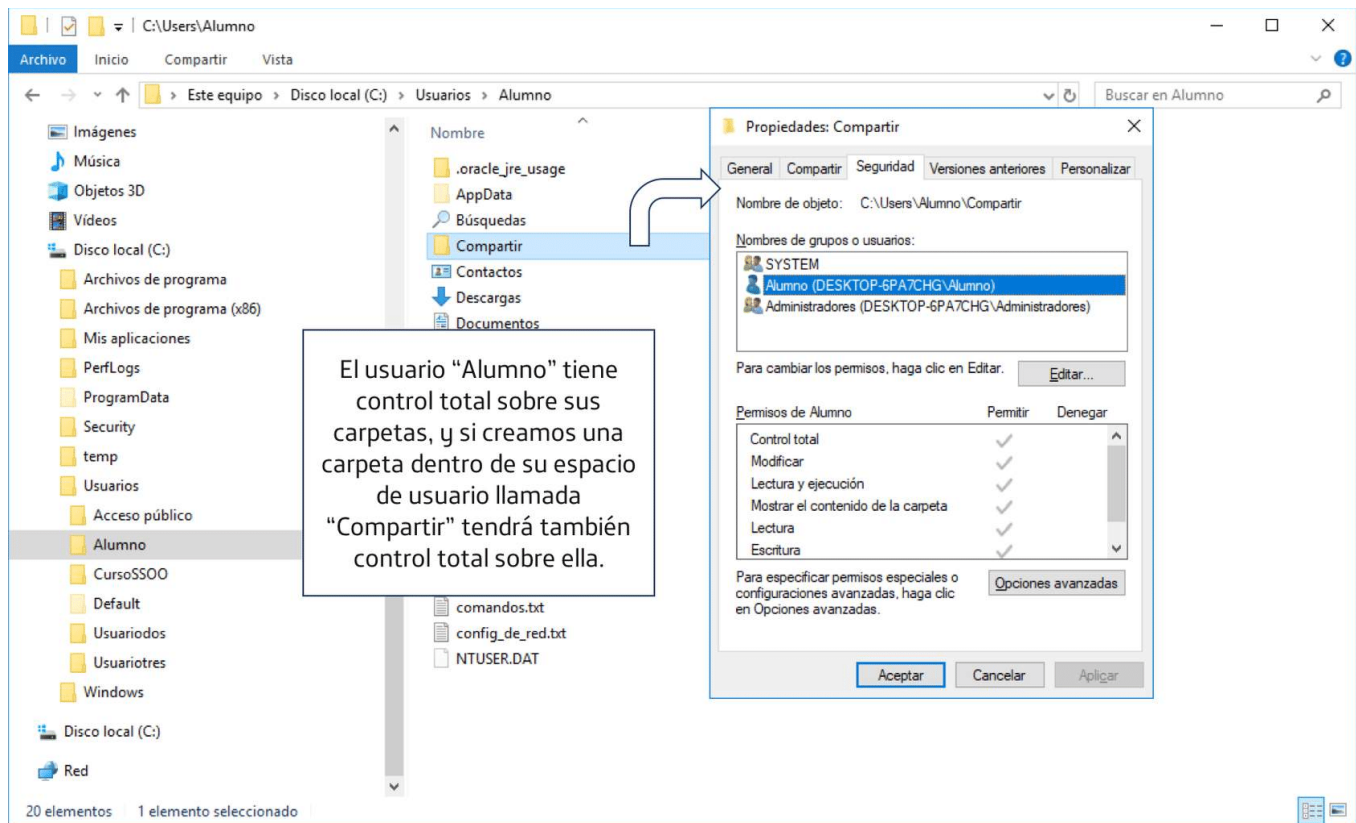
Para que veas cómo se "heredan" perfiles de permisos en Windows 10 te lo mostraremos en una secuencia de imágenes a continuación. Debes tener en cuenta que el comportamiento de cada sistema (p. ej. Linux u otras versiones del S.O.) puede ser muy diferente.

Lo que vamos a hacer es crear varios directorios dentro de un usuario del sistema y otorgarle permisos sobre ellos a los otros usuarios. Crearemos varios niveles de directorios y verás como se van transmitiendo los permisos, y que además podemos cambiar de "permisos heredados" y convertirlos en "permisos explícitos" para una carpeta.



Estamos en el sistema como usuario "Alumno", que es miembro del grupo de administradores, junto con "CursoSS00".

Además tenemos otros dos usuarios ("Usuariodos" y "Usuariotres") que solo son miembros del grupo "usuarios".



En nuestro ejemplo creamos dos subdirectorios dentro de "Compartir", y a uno le vamos a dejar la característica de herencia y al otro se la quitaremos.

De momento vemos que ambos directorios han heredado los mismos permisos que la carpeta a la que pertenecen ("Compartir").

2 elementos

Propiedades: DirCON_herencia

Nombre de objeto: C:\Users\Alumno\Compartir\DirCON_herencia

Nombres de grupos o usuarios:

- SYSTEM
- Alumno (DESKTOP-6PA7CHG\Alumno)
- Usuariodos (DESKTOP-6PA7CHG\Usuariodos)**
- Administradores (DESKTOP-6PA7CHG\Administradores)

Para cambiar los permisos, haga clic en Editar.

Permisos de Usuariodos

Permitir	Denegar
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>

Propiedades: DirSIN_herencia

Nombre de objeto: C:\Users\Alumno\Compartir\DirSIN_herencia

Nombres de grupos o usuarios:

- SYSTEM
- Alumno (DESKTOP-6PA7CHG\Alumno)
- Usuariotres (DESKTOP-6PA7CHG\Usuariotres)**
- Administradores (DESKTOP-6PA7CHG\Administradores)

Para cambiar los permisos, haga clic en Editar.

Permisos de Usuariotres

Permitir	Denegar
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>

A la carpeta "DirCONherencia" le añadimos permisos para "Usuariodos".

A la carpeta "DirSINherencia" le añadimos permisos para "Usuariotres".

1 elemento

Propiedades: HijoCONherencia

Nombre de objeto: C:\Users\Alumno\Compartir\DirCON_herencia\HijoCONherencia

Nombres de grupos o usuarios:

- SYSTEM
- Alumno (DESKTOP-6PA7CHG\Alumno)
- Usuariodos (DESKTOP-6PA7CHG\Usuariodos)**
- Administradores (DESKTOP-6PA7CHG\Administradores)

Para cambiar los permisos, haga clic en Editar.

Permisos de Usuariodos

Permitir	Denegar
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>

Configuración de seguridad avanzada para HijoCONherencia

Nombre: C:\Users\Alumno\Compartir\DirCON_herencia\HijoCONherencia

Propietario: Alumno (DESKTOP-6PA7CHG\Alumno)

Permisos

Tipo	Entidad de seguridad	Acceso	Hereditado de	Se aplica a
Permitir	Usuariodos (DESKTOP-6PA7CHG\Usuariodos)	Lectura, escritura y ejecución	C:\Users\Alumno\Compartir\DirCON_herencia\	Esta carpeta, subcarpetas y archivos
Permitir	SYSTEM	Control total	C:\Users\Alumno\Compartir\DirCON_herencia\	Esta carpeta, subcarpetas y archivos
Permitir	Administradores (DESKTOP-6PA7CHG\Administradores)	Control total	C:\Users\Alumno\Compartir\DirCON_herencia\	Esta carpeta, subcarpetas y archivos
Permitir	Alumno (DESKTOP-6PA7CHG\Alumno)	Control total	C:\Users\Alumno\Compartir\DirCON_herencia\	Esta carpeta, subcarpetas y archivos

Si creamos otra carpeta (HijoCONherencia) dentro del subdirectorio que tiene la característica de heredar los permisos, vemos que se crea el mismo perfil de permisos que su "padre", y en las opciones avanzadas además nos dice de quien hereda cada permiso (algunos vienen de "abuelos", "bisabuelos", etc., es decir, de las carpetas superiores que las contienen y que los han ido pasando a las creadas dentro de ellas).

Ahora veamos como trabajar para que NO se herede un permiso si no deseamos que se transmita a los subdirectorios dentro de otro.

Propiedades: DirSIN_herencia

General Compartir Seguridad Versiones anteriores Personalizar

Nombre de objeto: C:\Users\Alumno\Compartir\DirSIN_herencia

Nombres de grupos o usuarios:

- SYSTEM
- Alumno (DESKTOP-6PA7CHG\Alumno)
- Usuanotres (DESKTOP-6PA7CHG\Usuanotres)
- Administradores (DESKTOP-6PA7CHG\Administradores)

Para cambiar los permisos, haga clic en Editar.

Editar...

Permisos de SYSTEM

Permitir	Denegar
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>

Para especificar permisos especiales o configuraciones avanzadas, haga clic en Opciones avanzadas.

Opciones avanzadas

Aceptar Cancelar Aplicar

Configuración de seguridad avanzada para DirSIN_herencia

Nombre: C:\Users\Alumno\Compartir\DirSIN_herencia

Propietario: Alumno (DESKTOP-6PA7CHG\Alumno) Cambiar

Permisos Auditoría Acceso efectivo

Para obtener información adicional, haga doble clic en una entrada de permiso. Para modificar una entrada de permiso, seleccione la entrada y haga clic en Editar (si está disponible).

Entradas de permiso:

Tipo	Entidad de seguridad	Acceso	Heredada de	Se aplica a
Permitir	Usuanotres (DESKTOP-6PA7CHG\Usuanotres)	Lectura, escritura y ejecu...	Ninguno	Esta carpeta, subc...
Permitir	SYSTEM	Control total	C:\Users\Alumno\	Esta carpeta, subc...
Permitir	Administradores (DESKTOP-6PA7CHG\Administradores)	Control total	C:\Users\Alumno\	Esta carpeta, subc...
Permitir	Alumno (DESKTOP-6PA7CHG\Alumno)	Control total	C:\Users\Alumno\	Esta carpeta, subc...

Agregar Quitar Ver

Deshabilitar herencia

Vamos a quitarle la herencia a la carpeta "DirDINherencia"...

Reemplazar todas las entradas de permisos de objetos secundarios por entradas de permisos heredables de este objeto

Aceptar Cancelar Aplicar

Configuración de seguridad avanzada para DirSIN_herencia

Nombre: C:\Users\Alumno\Compartir\DirSIN_herencia

Propietario: Alumno (DESKTOP-6PA7CHG\Alumno) Cambiar

Permisos Auditoría Acceso efectivo

Para obtener información adicional, haga clic en Editar (si está disponible).

Entradas de permiso:

Tipo	Entidad de seguridad	Acceso	Heredada de	Se aplica a
Permitir	Usuanotres (DESKTOP-6PA7CHG\Usuanotres)	Lectura, escritura y ejecu...	Ninguno	Esta carpeta, subc...
Permitir	SYSTEM	Control total	C:\Users\Alumno\	Esta carpeta, subc...
Permitir	Administradores (DESKTOP-6PA7CHG\Administradores)	Control total	C:\Users\Alumno\	Esta carpeta, subc...
Permitir	Alumno (DESKTOP-6PA7CHG\Alumno)	Control total	C:\Users\Alumno\	Esta carpeta, subc...

Agregar Quitar Ver

Deshabilitar herencia

Reemplazar todas las entradas de permisos de objetos secundarios por entradas de permisos heredables de este objeto

Aceptar Cancelar Aplicar

Bloquear herencia

¿Qué desea hacer con los permisos heredados actuales?

Está a punto de bloquear la herencia en este objeto, lo que significa que los permisos heredados de un objeto primario ya no se aplicarán a este objeto.

→ Convertir los permisos heredados en permisos explícitos en este objeto.

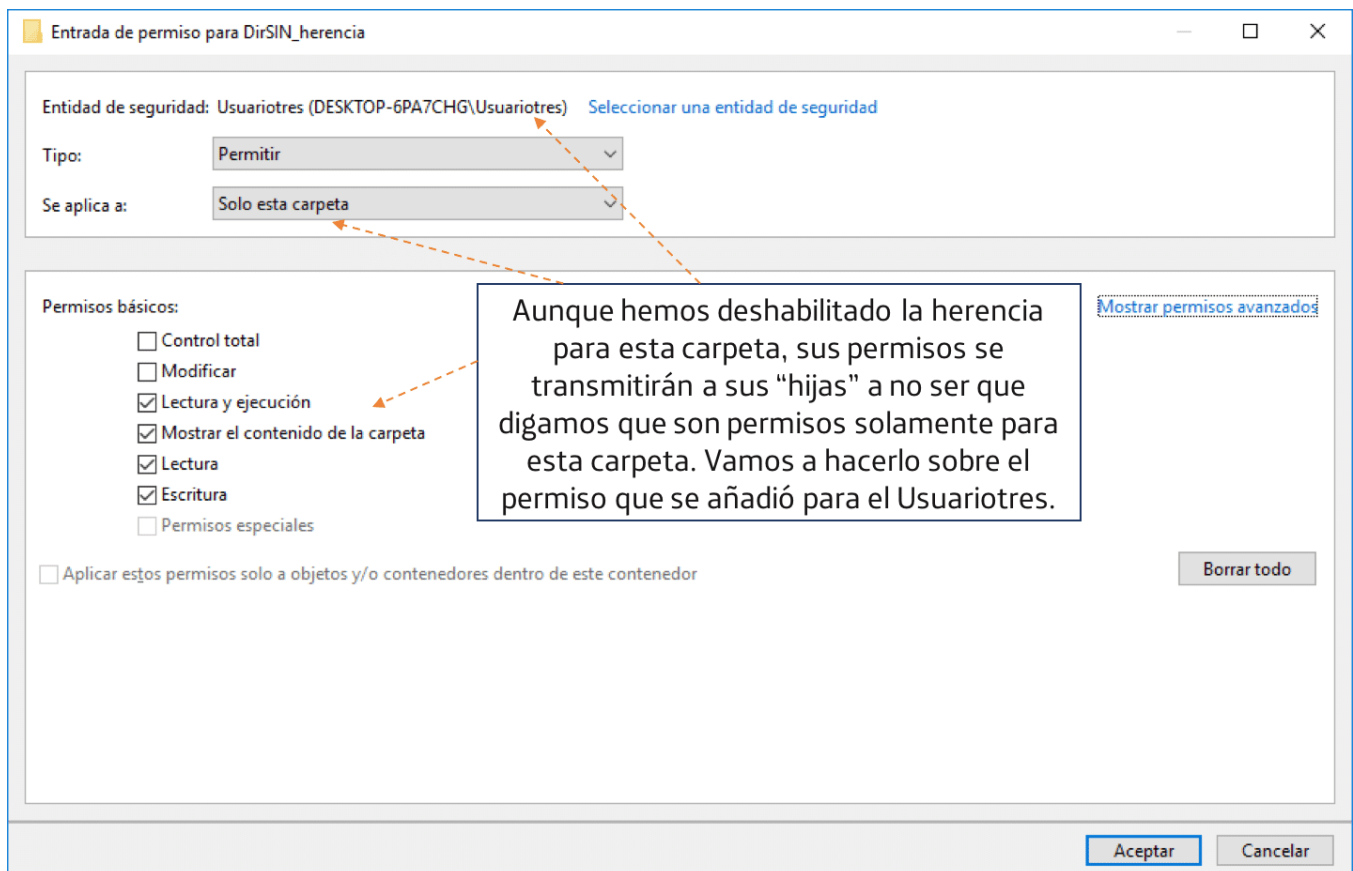
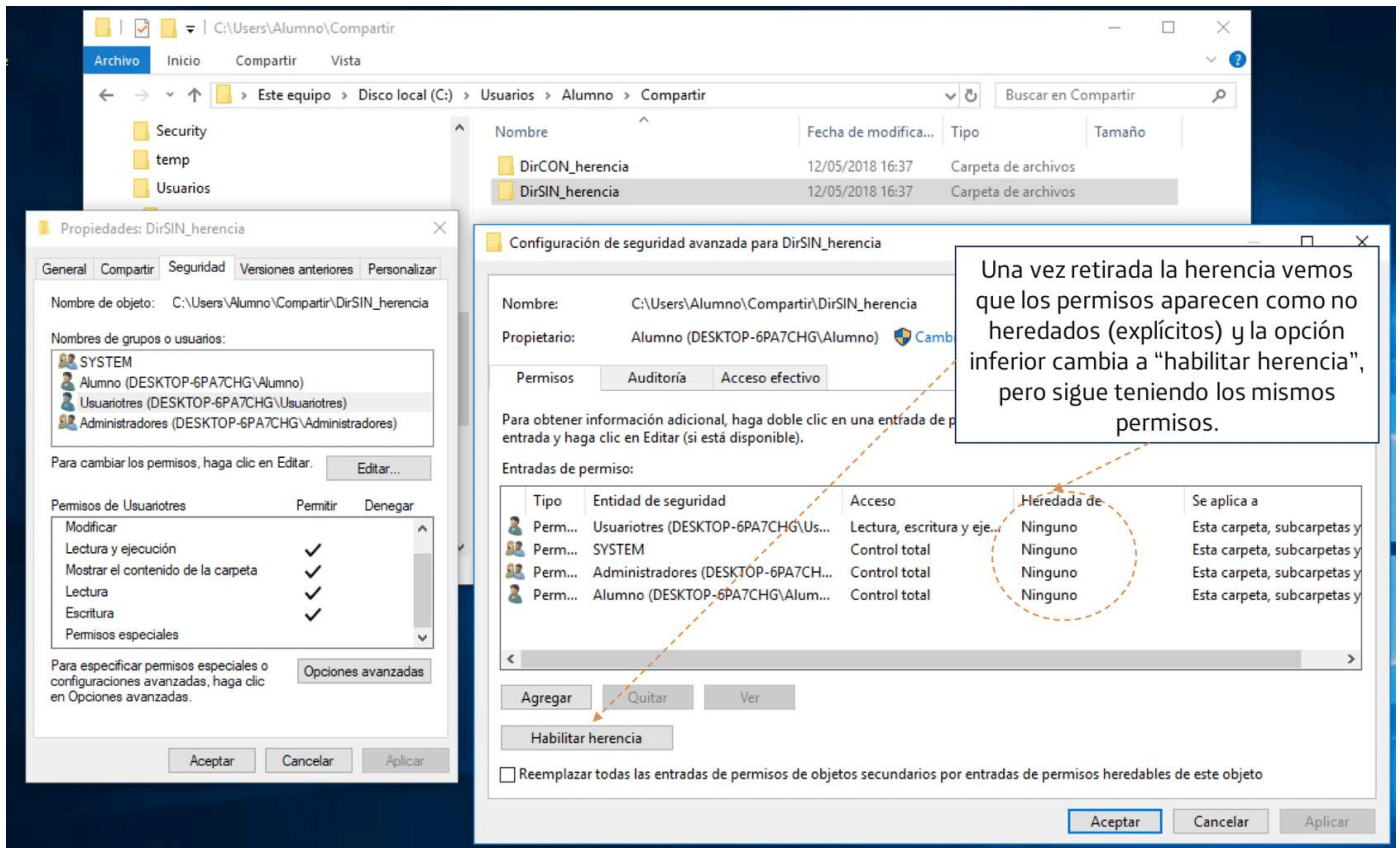
→ Quitar todos los permisos heredados de este objeto.

Cancelar

Al dar a la opción de "deshabilitar herencia" nos pregunta que queremos hacer con los permisos que tiene la carpeta.

Si elegimos la opción de quitar los permisos heredados la carpeta se quedaría solamente con los permisos explícitos que le hayamos configurado.

En este ejemplo elegiremos convertir todos los permisos en explícitos, y luego veremos si son heredados por sus carpetas "hijas" (las que hagamos dentro de ella).



The screenshot shows a Windows File Explorer window with the address bar set to 'C:\Users\Alumno\Compartir\DirSIN_herencia'. The left pane shows the 'Usuarios' folder. The right pane shows the 'HijaSinherenciatotal' folder. A context menu is open, showing the 'Propiedades' (Properties) option. The 'Seguridad' (Security) tab is selected, showing the 'Configuración de seguridad avanzada para HijaSinherenciatotal' dialog box. The 'Permisos' (Permissions) tab is active, showing a list of permissions for the folder. The 'Tipo' (Type) is 'Permitir' (Allow). The 'Entidad de seguridad' (Security entity) is 'Alumno (DESKTOP-6PA7CHG\Alumno)'. The 'Acceso' (Access) is 'Control total' (Full control). The 'Heredada de' (Inherited from) is 'C:\Users\Alumno\Compartir\DirSIN_herencia\'. The 'Se aplica a' (Applies to) is 'Esta carpeta, subcarpetas y archivos' (This folder, subfolders and files). The 'Reemplazar todas las entradas de permisos de objetos secundarios por entradas de permisos heredables de este objeto' checkbox is checked.

En efecto, vemos como la carpeta que hemos creado dentro de "DirSINherencia", llamada "HijaSinherenciatotal", se configura automáticamente con los permisos que están configurados como "heredables" para las subcarpetas dentro de ella, pero no con el permiso que marcamos como "solo para esta carpeta" (el de "Usuariotres")

Un último detalle: para poder configurar que un permiso sea "solo para esta carpeta" y no sea heredado por sus hijas, a su vez tiene que ser un permiso explícito y no heredado de "sus abuelas" (= carpetas de orden superior). Es decir, los directorios en Windows son muy cuidadosos de que sus permisos se transmitan como herencia a sus descendientes ☺.



Como resumen debes recordar que los permisos en Windows se transmiten a los subdirectorios y carpetas que creamos dentro de otra, salvo que los configuremos para que se apliquen exclusivos de esa carpeta y no de las subcarpetas que contenga.

Conflictos entre permisos

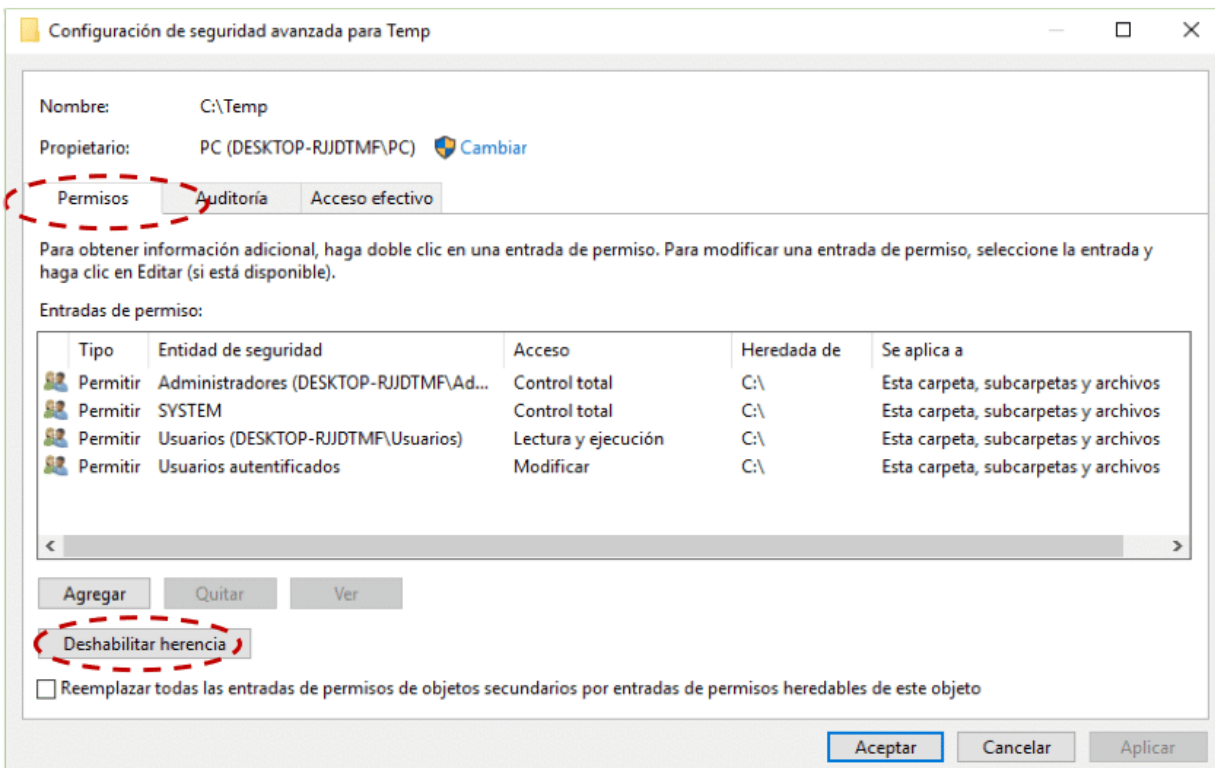
La gestión de los permisos sobre los objetos del sistema puede ser muy simple o llegar a tener una gran complejidad.

Al poder definirse para grupos y usuarios individuales, de forma local y en red, es posible que surjan **conflictos entre los diferentes permisos**.

Como hemos comentado, en general **los permisos asignados de forma explícita prevalecen sobre los permisos heredados**. Además, si una carpeta es compartida para todos con ciertos permisos y el usuario tiene otros como miembro de un grupo, los permisos se combinarán y en general se impone el más restrictivo, aunque no siempre.

Por ejemplo, si una carpeta está compartida en modo solo lectura pero el usuario pertenece a un grupo con control total (por ejemplo el de administración), entonces podrá tener control total sobre la carpeta y sus archivos.

Normalmente los **permisos los establece el propietario del archivo o el directorio, y solo puede cambiarlos el mismo usuario u otro que haya recibido permiso para ello**.



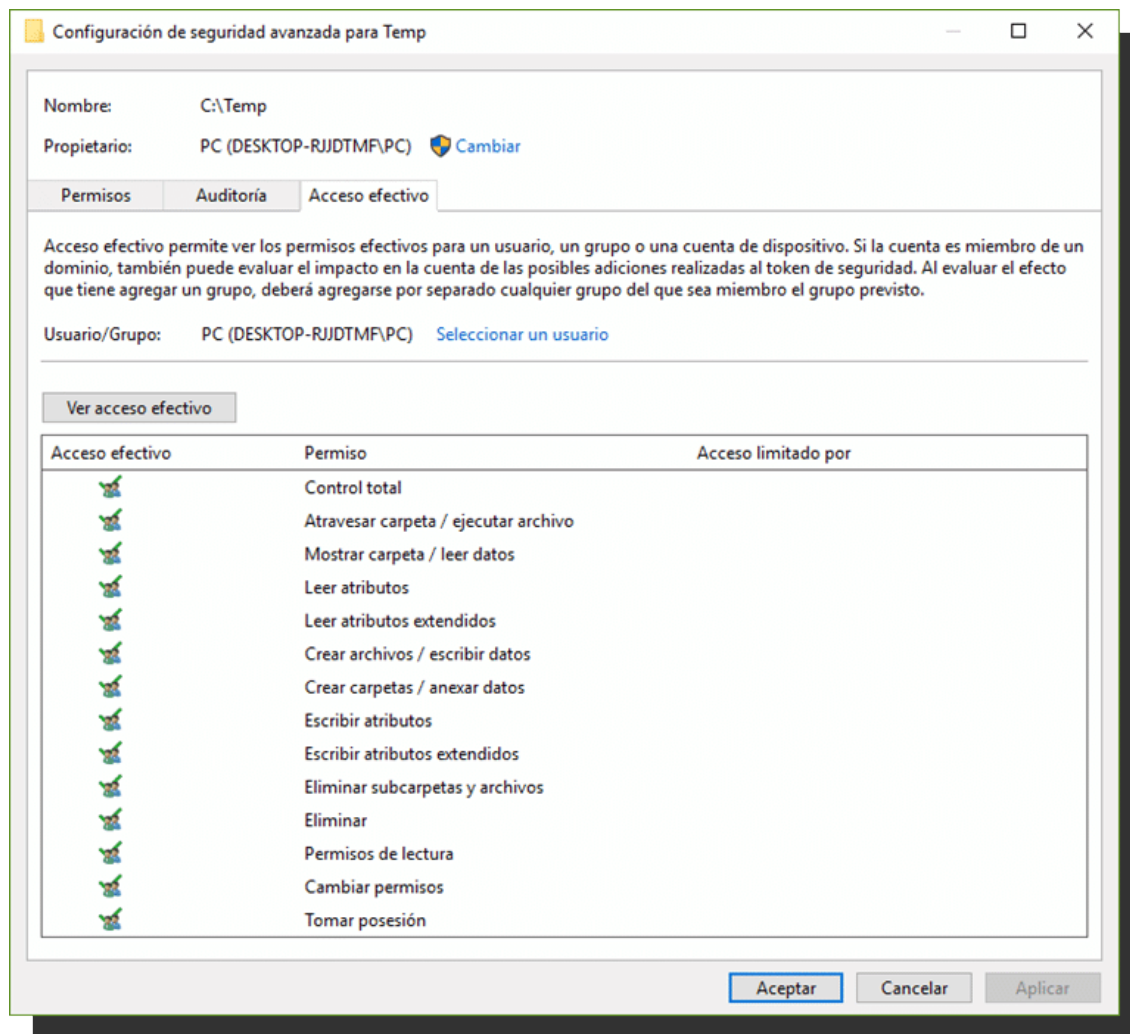
Cuando el permiso se establece para un directorio afectará a los ficheros y subdirectorios que dependen de él, tanto a los que ya existan (herencia) como a los que se creen posteriormente.

Permisos efectivos

Los permisos efectivos de un usuario sobre un determinado objeto son el resultado de combinar sus permisos individuales con los que puede tener como miembro de uno o varios grupos.

Por ejemplo, los **factores que se emplean (en Windows) para determinar los permisos efectivos** son:

- Pertenencia al grupo global.
- Pertenencia al grupo local.
- Permisos locales.
- Privilegios locales.
- Pertenencia a grupos universales.



Características de los permisos efectivos

Lo que debes recordar sobre las características de los **permisos efectivos** es lo siguiente:

- Son una **combinación de los permisos** concedidos al usuario y a todos los grupos a los que pertenece.
- Los permisos de archivos prevalecen sobre los de carpeta.
- La “denegación” sobrescribe a otros permisos.
- Normalmente el propietario del objeto controla cómo se configuran los permisos del objeto y a quién se conceden. Por otra parte, el propietario puede ser de forma predeterminada el grupo de administradores.

Delegación de permisos

Delegar un permiso quiere decir que un usuario que lo posee puede cederlo (delegarlo) a otro usuario.

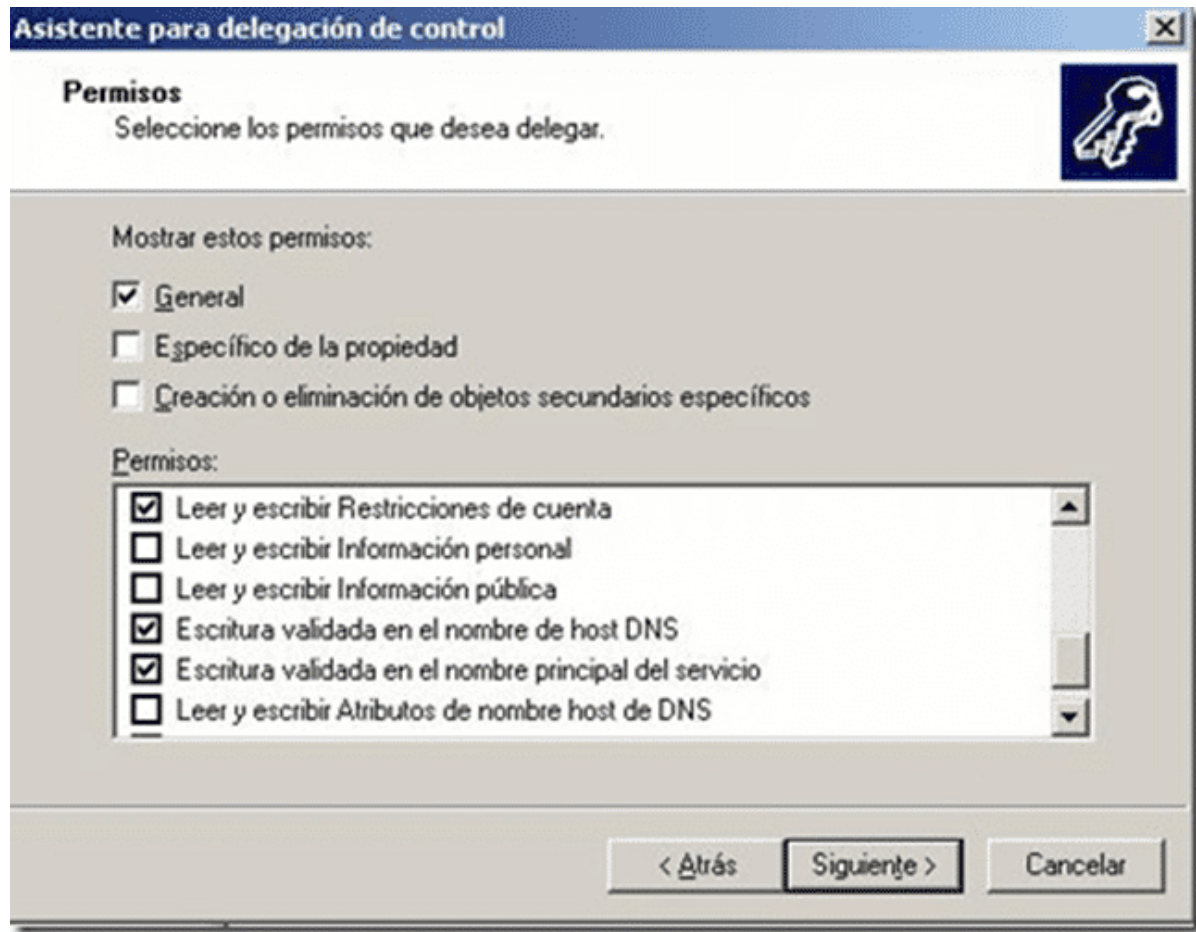
Para poder hacerlo, el usuario propietario tiene a su vez que tener el **permiso de delegación** sobre el que quiere delegar, es decir, estar autorizado a delegarlo.

La delegación de permisos va más allá de una simple función de “dar acceso a...”, y es un elemento que puede facilitar mucho la división de tareas de administración en un sistema, aunque en sistemas diferentes los procedimientos empleados para hacerlo y la terminología asociada puedan ser diferentes.

Recuerda

No es lo mismo la delegación de un permiso que “asumir el rol de otro usuario”. En algunos sistemas un usuario puede asumir el rol de otro usuario a través, por ejemplo, de un comando (como el caso de “su” en Unix), de forma que el usuario asume los atributos del rol que adquiere y “pierde” los suyos. Pueden existir roles predefinidos, como por ejemplo “administrador principal”, “administrador del sistema”, operador, etc.

Tampoco es lo mismo que la delegación de tareas o funciones. Por ejemplo, cuando un usuario de un grupo de administración puede delegar sobre otros ciertas tareas como la creación de cuentas de usuario, ello sin hacerle pertenecer al grupo de administración.



Sin embargo, ambas cosas están relacionadas, pues **para poder delegar tareas que impliquen permisos sobre un objeto, el usuario debe también tener delegados los permisos correspondientes**. Por ejemplo, si delegamos la tarea de crear usuarios sobre un usuario/grupo, este podrá crear nuevos usuarios en el sistema, pero si intenta incluirlos en un grupo sobre el que no tiene permisos, el sistema no le dejará.

Ejemplo

En algunos sistemas (por ejemplo en Windows Server 2008 R2) **se pueden delegar permisos administrativos de forma personalizada** sobre algún usuario o grupo, y la delegación de permisos se puede realizar a nivel de dominio completo o a nivel de unidades organizativas. Dentro de una "unidad organizativa" podemos tener varias cuentas de usuario, recursos compartidos, etc.

Esta posibilidad permite evitar tener varios administradores con amplios permisos administrativos, ya que podemos delegar solamente los permisos necesarios para una tarea sobre aquellos encargados de realizarla, y el hecho de poder limitar los permisos delegados a un dominio concreto es también un elemento de seguridad. Recordemos que un dominio constituye en sí mismo un límite de seguridad.



La **delegación de permisos** es bastante común en sistemas en red con una gran cantidad de equipos y usuarios a gestionar, por ello se suele ofrecer esta posibilidad en los sistemas operativos en red y no en los diseñados para equipos locales ("versiones *desktop* locales") como los que usamos nosotros. En este curso queremos que tengas el concepto de lo que significa "delegar un permiso", pero no entraremos en más complejidades. Recuerda sobre todo la diferencia entre delegar y asumir el rol de otro usuario, lo cual sí hemos visto con el comando "*su*" en Linux.

Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado:

En esta unidad has visto que no es lo mismo un "**permiso**" sobre un recurso del sistema que un "**derecho**" de un usuario que forma parte de su perfil de usuario. Además, te habrás dado cuenta de que la gestión de los permisos puede complicarse bastante en sistemas en red con un gran número de recursos y usuarios. Definir los perfiles de actuación y los grupos de usuarios que pueden hacerlo es muy importante para un administrador del sistema, ya que puede facilitarle su labor o, por el contrario, complicársela mucho si, tanto los permisos como los perfiles de usuario, no están bien definidos.

unir LA UNIVERSIDAD
EN INTERNET | FORMACIÓN
PROFESIONAL

PROEDUCA