1

SEGURIDAD EN LA PUBLICACIÓN DE PÁGINAS WEB

Cuando se habla de seguridad web existen algunas premisas que se deben tener en cuenta si se desea proteger un sitio o aplicación web. A continuación, se proporcionan algunas de las más importantes hoy en día:

- Utilizar un sistema de archivos seguro que garantice que la información se almacena de forma permanente, adecuada y segura de forma que no se pierda ningún dato ni información importante ante un corte o pérdida de suministro eléctrico.
- Realizar copias de seguridad periódicas y almacenarlas en dispositivos externos seguros, a ser posible desconectados de la red.
- Establecer y ejecutar las aplicaciones con un nivel de privilegios adecuado para garantizar la correcta gestión y manipulación de la información.
- Acceder a los datos de forma segura, ya sea a nivel de base de datos, como a nivel de lectura y escritura de archivos. Aquí también es importante tener en cuenta el tamaño del buffer y de las lecturas y escrituras que se realizan.
- Realizar una comunicación entre la máquina y el usuario de forma comprensible y segura sin presentar información sensible que pueda ser utilizada con fines delictivos o malintencionados.
- Mantener actualizadas con la última versión las herramientas y/o aplicaciones de desarrollo (IDEs, editores de texto o código, etcétera),

los plugins, frameworks y librerías usadas en los proyectos y las últimas revisiones de seguridad del sistema operativo que se esté usando.

- Utilizar un uso correcto de cookies, Session Storage, Local Storage y variables de sesión ya que pueden ser fruto de ataques de suplantación de identidad, SQL Injection u otros usos malintencionados. Esto, por supuesto, implica que no se debe almacenar información sensible a no ser que sea vital o necesario y esté totalmente justificado a nivel lógico y legislativo (por la RGPD).
- Si se ofrece la posibilidad de subir archivos se deben establecer unos correctos límites de tamaño y de tipo para evitar malos comportamientos y usos indebidos.

1.1 GLOSARIO DE TÉRMINOS

1.1.1 Qué es un inodo

Un inodo, nodo-i, o nodo índice, es una estructura de datos propia de los sistemas de archivos, los cuales se verán a continuación.

Los inodos es frecuente encontrarlos en los sistemas operativos de tipo Unix o Linux y, normalmente, contienen las características de lo que comúnmente denominamos archivo o directorio, aunque también puede ser cualquier otro objeto que pueda contener el sistema de archivos como un enlace simbólico.

Para entender un poco qué son y cómo funcionan pensemos que cada inodo es reconocido por un número entero y único dentro del sistema. Los directorios, por su parte, recuperan una lista de parejas formadas por un número de inodo y nombre identificativo que permite el acceso al archivo en cuestión. Por ello, aunque cada archivo tiene un único inodo, puede haber más de un nombre en distintos o incluso en el mismo directorio, lo que facilita su localización.

1.1.2 Diferencia entre clúster y sector

Cuando hablamos de sistemas de archivos hay que tener claros dos conceptos. Por un lado, lo que es un clúster y, por otro, lo que es un sector.

Mientras que un sector es la unidad lógica más pequeña que identifica un espacio en disco (habitualmente desde 512B hasta 128KB), el clúster es el conjunto contiguo de sectores que componen la unidad más pequeña para almacenar archivos, directorios y/o cualquier otro objeto que pueda contener el sistema de archivos.

1.1.3 Unidades del sistema digital

Las unidades del sistema digital podríamos definirlas como aquellas unidades de medición que permiten determinar el valor decimal de una entidad digital a partir de su valor binario. Aunque existen más medidas, aquí mostramos las más comunes hoy en día:

Medida	Simbología	Equivalencia	Valor en bytes
Byte	B	8 bits	1 B
Kilobyte	KB	1024 B	1.024 B
Megabyte	MB	1024 KB	1.048.576 B
Gigabyte	GB	1024 MB	1.073.741.824 B
Terabyte	TB	1024 GB	1.099.511.627.776 B
Petabyte	PB	1024 TB	1.125.899.906.842.624 B
Exabyte	EB	1024 PB	1.152.921.504.606.847.000 B
Zetabyte	ZB	1024 EB	1.180.591.620.717.411.303.424 B
Yottabyte	YB	1024 ZB	1.208.925.819.614.629.174.706.176 B

1.2 DEFINICIÓN DE ARCHIVO Y DIRECTORIO

Un archivo podría definirse como un conjunto de información o secuencia de bytes que se encuentran almacenados en un dispositivo y que viene identificado por un nombre y la descripción de la carpeta o directorio que lo contiene.

Un directorio, por su parte, no es más que un contenedor virtual en el que se encuentran uno o varios archivos y otros contenedores dependientes de él denominados subdirectorios. Su relevancia no es una cuestión de broma pues almacenan información importante como los atributos de cada uno de los archivos o dónde se encuentran físicamente en el dispositivo de almacenamiento.

1.3 DEFINICIÓN DE SISTEMA DE ARCHIVOS

Un sistema de archivos o sistema de ficheros (en inglés File System), es un elemento o componente que controla cómo se almacenan y recuperan los datos en dispositivos de almacenamiento interno y externo.

Una de las razones más importantes, si no la más, de usar un sistema de archivos es básicamente para poder determinar dónde termina un dato o archivo y dónde comienza el siguiente. Además, es el encargado de administrar y acceder a la información que se almacena de forma persistente.

Entre sus principales funciones podemos encontrar la asignación de espacio a cada archivo, la administración del espacio libre, el control y acceso a los datos almacenados y proporcionar una forma de estructurar la información guardada en dispositivos o unidades de almacenamiento (normalmente discos duros o unidades de estado sólido), con el fin de ser representada textual o gráficamente usando una herramienta denominada gestor de archivos.

En lo referente a la asignación de espacio de los archivos y la administración del espacio libre, lo habitual es utilizar cadenas de bloques de tamaño fijo (también denominados sectores) y que, frecuentemente son de 512 bytes de longitud (también denominados clústeres). Una vez que se han definido estas entidades, el sistema de archivos podrá ser capaz de organizar los archivos y directorios y mantener un registro de qué sectores y clústeres pertenecen a cada archivo y, cómo no, cuáles no han sido utilizados, lo que nos permitirá conocer cuáles están libres y cuánto espacio libre queda.

En lo referente al control y acceso a los datos almacenados los sistemas de archivos nos proporcionan una forma para crear, editar, mover, renombrar y eliminar la información, tanto a nivel de archivo, como a nivel de directorio, pero carecen de métodos para crear enlaces adicionales a un directorio o archivo y renombrar a los enlaces de niveles superiores (habitualmente reconocidos por "..").

Como dato curioso para los que no lo sepan, cuando se elimina un archivo de un dispositivo de almacenamiento persistente, en realidad, lo que se está haciendo es una desvinculación del registro o tabla de archivos, lo que significa que no está visible ni accesible, pero seguirá ahí. No obstante, sus sectores y clústeres serán considerados como disponibles, por lo que podrá ser utilizado por otro archivo u archivos posteriormente.

Como dato final hay que destacar que los sistemas de archivos también pueden ser utilizados para acceder a datos generados dinámicamente, como los recibidos a través de una conexión de red, es decir, sin la intervención de un dispositivo de almacenamiento interno o externo.

1.3.1 Sistema de archivos Linux

Linux es el nombre de un tipo de sistema operativo muy ligero, multiusuario, multitarea y multiplataforma basado en Unix que se encuentra bajo la licencia GNU GPL (General Public License o Licencia Pública General de GNU).

La mayoría de los sistemas Linux son gratuitos y proveen de todo lo necesario para poder trabajar en un ordenador, al igual que sucede con otros sistemas operativos, sin embargo, una de sus características más importantes es que puede ser utilizado, copiado, modificado y/o redistribuido libremente para cualquier uso que se desee darle, siempre que no incumpla los términos de la licencia GPL de GNU.

En lo referente al sistema de archivos que maneja, Linux resulta ser el más versátil y compatible ya que permite, desde el uso de sistemas basados en discos, como puedan ser ext4, ReiserFS, FAT32 o NTFS, hasta sistemas de ficheros de comunicación en red, como NFS o SMB.

1.3.1.1 EXTENDED FILE SYSTEM

El sistema de archivos extendido (Extended File System o ext), fue el primer sistema de archivos creado específicamente para el sistema operativo Linux. Fue el primero que usó el sistema de archivos virtual y dio la capacidad de manejar sistemas de almacenamiento de hasta 2GB.

Sin embargo, el sistema de archivos ext no fue nada más que la base del iceberg pues, detrás de él, se desarrollaron nuevas y mejores versiones.

1.3.1.1.1 Second Extended File System

El segundo sistema de archivos extendido o ext2 es la evolución lógica del sistema ext, aunque actualmente se encuentra un poco en desuso.

Entre sus principales cualidades podemos destacar que es un sistema con una gran estabilidad, que posee una fragmentación muy baja, permite manejar sistemas de archivos de hasta 4TB y proporciona nombres de ficheros de hasta 255 caracteres con un tamaño máximo de hasta 2GB, aunque se vuelve algo lento cuanto mayor es el tamaño de los archivos.

1.3.1.1.2 Third Extended File System

El tercer sistema de archivos extendido o ext3 es una mejora del sistema ext2 que posee un registro diario y es totalmente compatible con su predecesor.

Entre sus principales cualidades podemos destacar que posee una alta fiabilidad, una gran capacidad de actualización ya que permite actualizarse a ext3 incluso si el sistema ext2 está montado y contempla la previsión de pérdida de datos por fallos del disco o apagones. Sin embargo, presenta la imposibilidad de recuperar datos borrados.

En lo referente a sus límites podemos destacar que ext3 presenta diferentes valores en función del tamaño del bloque, archivo y sistema.

Tamaño del bloque	TamMáx. de archivo	TamMáx. del sistema de archivos
1 KB	16 GB	2 TB
2 KB	256 GB	8 TB
4 KB	2 TB	16 TB
8 KB	2 TB	32 TB

1.3.1.1.3 Fourth Extended File System

El cuarto sistema de archivos extendido o ext4 es un sistema de archivos transaccional que resulta ser una mejora del ext3.

Entre sus principales cualidades podemos destacar que hace un mejor uso de CPU, incrementa la velocidad de lectura y escritura, amplia los límites de tamaño de archivos hasta 16TB y desarrolla un considerable aumento de la capacidad del sistema de ficheros ya que puede llegar a los 1024PB (1 PetaByte es equivalente a 1024TB).

1.3.1.2 REISER FILE SYSTEM

Reiser File System es un sistema de ficheros de última generación y propósito general que permite organizar los ficheros de modo que se puedan optimizar y agilizar las operaciones con estos. Sin embargo, puede presentar un problema ya que se está discutiendo su eliminación del kernel de Linux desde principios de 2022 debido a la falta de soporte, mantenimiento y errores técnicos inherentes al sistema de archivos.

1.3.1.3 JOURNALED FILE SYSTEM

Journaled File System o JFS es un sistema de archivos de 64-bit con respaldo de transacciones creado por IBM que cuenta con diversas versiones en función de si nos encontramos en un sistema AIX, eComStation, OS/2, Linux o HP-UX.

Entre sus principales cualidades podemos destacar que posee un muy eficiente respaldo de transacciones (Journaling) y de la administración de directorios y que mejora el uso de la memoria mediante adjudicación dinámica de Inodos.

1.3.1.4 XFS

XFS es un sistema de archivos de 64 bits con registro de bitácora de alto rendimiento creado por Silicon Graphics Inc. cuyo objetivo era su implementación en sistemas UNIX, aunque en mayo de 2000 fue liberado bajo una licencia de código abierto.

Entre sus principales cualidades podemos destacar que admite un sistema de archivos que va desde los 16TB hasta 8EB (8 exabytes), dependiendo de si el sistema operativo es de 32 o 64 bits.

1.3.1.5 SWAP

Swap es un sistema de ficheros usado comúnmente para la partición de intercambio de Linux. Esto se debe a que todos los sistemas Linux necesitan de una partición de este tipo para cargar los programas y no saturar la memoria RAM cuando se excede su capacidad.

En Windows, sin embargo, esto se hace con el archivo pagefile.sys en la misma partición de trabajo, con los problemas que esto conlleva.

1.3.1.6 NETWORK FILE SYSTEM

Network File System o NFS es un sistema de archivos en red que posibilita acceder a recursos remotos como si fuesen locales. Fue desarrollado inicialmente por Sun Microsystems y es una opción bastante estándar en sistemas de ficheros en red que corren sobre GNU/Linux.

1.3.1.7 COMMON INTERNET FILE SYSTEM

Network File System o CIFS, a veces denominado SMB o Samba, es otro sistema de archivos en red que posibilita acceder a recursos remotos como si fuesen locales, no obstante, su objetivo es muy distinto en función de cómo y quién lo utilice.

Mientras que en Windows se suele usar más para compartir ficheros e impresoras por la red, en Linux se suele usar para permitir la convivencia simultánea de sistemas Windows y GNU/Linux en la misma red de área local.

1.3.1.8 SYSFS

SysFS es otro sistema de archivos virtual que, básicamente, exporta información sobre los dispositivos del sistema (hardware), sus controladores hacia el espacio del usuario y permite configurar alguno de sus parámetros.

1.3.2 Sistemas de archivos Windows

Windows es el nombre es un tipo de sistema operativo mucho menos ligero que Linux, pero que también resulta ser multiusuario, multitarea y está disponible para arquitecturas x86, x64 y ARM. También se suele identificar con el nombre de una familia de distribuciones para ordenadores y servidores desarrollados y vendidos por la compañía Microsoft.

En lo referente al sistema de archivos que maneja, Windows permite, básicamente FAT, FAT32, NTFS. A continuación, se comentan cada uno de ellos.

1.3.2.1 FILE ALLOCATION TABLE

El sistema de archivos File Allocation Table, FAT o Tabla de Asignación de Archivos, comenzó a utilizarse en MS-DOS allá por 1980 y fue considerado como sistema de archivos principal de las distribuciones no empresariales de Microsoft Windows hasta Windows ME.

Sus versiones reciben el nombre de su acrónimo y un valor que hace referencia número de bits y su característica más llamativa es que los archivos ocupan los clústeres enteros, es decir, si un archivo ocupa 36KB y el tamaño del clúster es de 32KB, se estarán ocupando 2 clústeres que, en espacio real, será como si el archivo ocupase 64KB.

No obstante, el sistema de archivos FAT sigue siendo muy frecuente ya que se suele utilizar en sistemas de almacenamiento extraíbles como memorias SD o USB y en memorias internas de dispositivos como cámaras digitales, smartphones o televisores.

1.3.2.1.1 FAT16

El sistema de archivos FAT16 es un sistema de 16 bits que permite el nombrado e identificación de nombres con hasta 12 caracteres, 8 para el nombre, 1 para el símbolo punto y 3 para la extensión.

Al ser un sistema de 16 bits el tamaño máximo de un archivo se estableció en 2GB con un número máximo de 65.536 clústeres de 32KB cada uno, lo que hace que se puedan gestionar particiones desde 256MB hasta los 2GB.

1.3.2.1.2 FAT32

El sistema de archivos FAT32 fue la evolución natural del sistema FAT16 y, como dato curioso, diremos que se dio a conocer a través del Windows 95 OSR2.

Aunque el sistema de archivos FAT32 utiliza 32 bits, sólo 28 de ellos se usan para las entradas ya que, los otros 4 se reservaron para el futuro. Esto explica que, aunque tenga 32 bits, al usar sólo 28, el tamaño máximo de un archivo se establezca en 4GB con un número máximo de 268.173.300 clústeres de 32KB cada uno y un tamaño máximo de partición de hasta 2TB.

Sin embargo, con esta evolutivo también aumentaron los problemas de fragmentación, lo que terminó por volver más lento el sistema y el acceso a los archivos.

1.3.2.2 EXTENDED FILE ALLOCATION TABLE

El sistema de archivos exFAT es otra evolución del sistema de archivos FAT32 que está disponible desde 2006 y que utiliza 64 bits, en vez de los 32 de su predecesor.

Cabe destacar que el objetivo de este nuevo sistema de archivos nunca fue mejorar al sistema de archivos FAT32, sino que se pensó para cubrir las necesidades de los sistemas o unidades de almacenamiento externos o extraíbles. Por esta razón es frecuente encontrarlo en memorias USB de gran tamaño y discos SSD.

Al ser un sistema de 64 bits el tamaño máximo de un archivo se estableció en 16EB con un tamaño de clúster de hasta 2^{255} bytes y un soporte de hasta 2.796.202 archivos y directorios por partición.

1.3.2.3 NEW TECHNOLOGY FILE SYSTEM

El sistema de archivos New Technology File System o NTFS es un sistema de archivos transaccional (Journaling) desarrollado por Microsoft que posee la capacidad de autocorrección y una mayor confiabilidad y seguridad en lo que a datos se refiere.

Entre sus principales cualidades podemos destacar que permite definir el tamaño del clúster a partir de 512 bytes (tamaño mínimo de un sector) de

forma independiente al tamaño de la partición, permite comprimir los medios de almacenamiento y soporta de 2^{32}–1 clústeres, que se traduce en aproximadamente 16TB usando clústeres de 4KB.

Además, también permite la indexación de datos, proporciona nombre y accesos más largos y ofrece una gran compatibilidad con unidades de almacenamiento de gran tamaño.

1.3.2.4 DIRECTORIOS IMPORTANTES

En lo referente a los directorios importantes que presenta Windows hay que llamar la atención sobre 3:

- **%USERPROFILE%**: es el directorio donde se almacenan los perfiles de usuario en Windows.
- **%WINDIR%**: es el directorio donde se encuentra instalado el sistema operativo Windows. En la mayoría de las ocasiones este directorio se identifica con el nombre del sistema operativo (Windows), aunque hay versiones donde el nombre de esta carpeta puede personalizarse durante el proceso de instalación. Dentro de la carpeta Windows podemos encontrar otras carpetas y archivos importantes que afectan a la configuración y servicios disponibles como el tipo de letra, el acceso a las direcciones IP con sus hosts, los controladores de todos y cada uno de los periféricos y componentes del PC, y la carpeta de archivos temporales. En ocasiones puede que encontremos carpetas vacías, pero nunca será una buena idea eliminarlas o cambiarlas de nombre pues puede que afecte al buen funcionamiento del sistema operativo o alguna de sus aplicaciones.
- **%PROGRAMFILES%**: es el directorio en donde se instalan las aplicaciones que no forman parte del propio sistema operativo. Aquí se suelen instalar también las aplicaciones de los usuarios y, en general, podemos decir que posee dos ubicaciones, una para los programas que funcionan con la arquitectura x86 y otra para los programas que funcionan con la arquitectura x64.

1.3.3 Sistemas de archivos Mac OS

MacOS es el nombre de una serie de sistemas operativos gráficos desarrollados y comercializados por Apple desde 1985. Al igual que Windows y Linux, permite el multiusuario y la multitarea, sin embargo, mientras que Linux y Windows son

bastante estables cuando se instalan en cualquier PC, su compatibilidad y estabilidad sólo está garantizada en equipos que sean de la propia compañía.

En lo referente al sistema de archivos que maneja, MacOS permite, básicamente HFS+ y APFS, el cual pasamos a describir a continuación:

1.3.3.1 HFS+

El sistema de archivos HFS+ es una evolución del HFS, el cual presentaba limitaciones similares al sistema de archivos FAT. Entre sus principales características podemos destacar que permite nombres de fichero de hasta 255 caracteres de longitud UTF-16, y que utiliza una tabla de asignación de 32 bits, en lugar de los 16 bits de HFS.

1.3.3.2 APFS

El sistema de archivos APFS es el reemplazo natural de HFS+ y es el sistema de archivos que actualmente usa Apple. Según informan en su página oficial utiliza números de inodo de 64 bits para poder ofrecer un almacenamiento mucho más seguro, permite una encriptación de alta seguridad, espacio compartido, instantáneas, redimensionamiento rápido de los directorios y mejora de los aspectos básicos del sistema de archivos.

1.3.4 Estándar FHS (Filesystem Hierarchy Standard)

El estándar de jerarquía del sistema de archivos (en inglés Filesystem Hierarchy Standard, también conocido por sus siglas FHS) se diseñó originalmente en 1994 y tenía como objetivo definir la estructura de directorios principales y sus contenidos en sistemas operativos GNU/Linux. Posteriormente fue utilizado e implantado en sistemas Mac OS y otros sistemas de la familia Unix.

Este estándar se basa en que todos los archivos y directorios son dependientes del directorio raíz, identificado por una barra inclinada hacia la derecha "/", aun cuando, éstos, se encuentren en distintos dispositivos físicos.

Cabe destacar que, aunque la mayoría de los directorios que el estándar FHS presenta están en los sistemas operativos que lo usan, sus descripciones pueden no coincidir exactamente con el originalmente definido ya que pueden no ser considerados obligatorios por algunas plataformas.

1.3.4.1 ESTRUCTURA DE LOS DIRECTORIOS GENERAL

En la estructura de directorios general se pueden dar varias subjerarquías de directorios con diversas funciones de almacenamiento y organización en todo el sistema. No obstante, estos directorios pueden ser clasificados en:

- **Estáticos**: aquellos directorios que contienen archivos que no son modificados sin la intervención del administrador (root), aunque puedan ser accedidos por cualquier otro usuario.
- **Dinámicos**: aquellos directorios que contienen archivos que pueden ser leídos, escritos y/o modificables por otros usuarios (si tienen permiso), aparte del administrador. Dentro de estos directorios lo frecuente es encontrar configuraciones, documentos, archivos de usuario...
- **Compartidos**: aquellos directorios que contienen archivos que se pueden encontrar en un sistema y utilizarse en otro y pueden ser compartidos entre los diferentes usuarios.
- **Restringidos**: aquellos directorios que contienen ficheros de uso privado, sistema y/o que no pueden ser compartidos, ya que sólo son manipulables por el administrador.

1.3.4.2 ESPECIFICANDO LOS DIRECTORIOS DEFINIDOS POR FHS

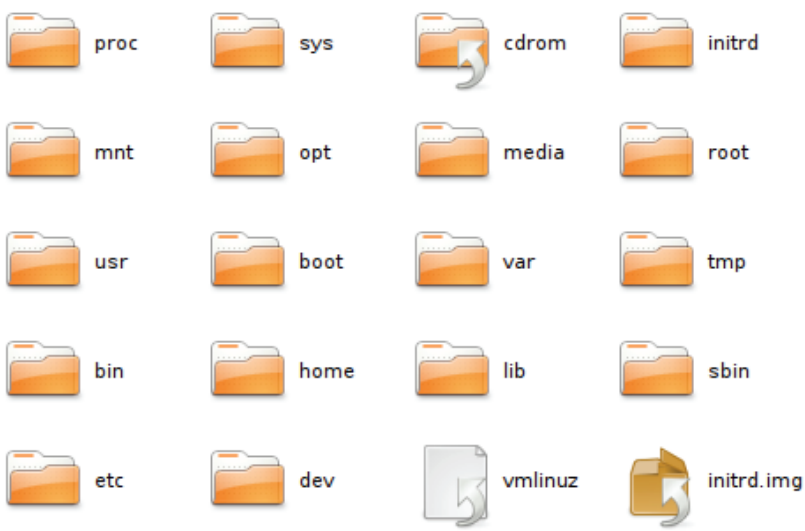


Imagen Sistema de archivos (Linux).png, extraída de Wikimedia Commons contributors

Directorio	Descripción Simple
/	Es el directorio raíz o contenedor de todo el sistema de jerarquía.
/bin/	Es el directorio con todos los comandos que son esenciales para el funcionamiento adecuado de las sesiones de usuario. Por ejemplo, aquí encontremos comandos como cat, ls, cp, rm, mkdir, etc.
/boot/	Es el directorio que contiene todos los archivos asociados al arranque del sistema (como los núcleos o el initrd), generalmente en una partición o disco aparte.
/dev/	Es el directorio que contiene los archivos especiales vinculados a dispositivos hardware.
/etc/	Es el directorio que contiene los archivos de configuración del sistema específicos del Host de todo el sistema. Su nombre se suele atribuir a Editable Text Configuration (Configuración de texto editable).
/etc/opt/	Es el directorio que contiene los archivos de configuración para los programas alojados dentro del directorio opt.
/etc/X11/	Es el directorio que contiene la configuración de las X Window System, versión 11.
/etc/sgml/ **/etc/xml/**	Son los directorios que contienen los archivos de configuración para SGML y XML, respectivamente.
/home/	Es el directorio que contiene las carpetas de trabajo de todos los usuarios, exceptuando el superusuario o root. En general, suele estar en una partición o sección separada y contiene los archivos guardados, ajustes personales, ... de cada usuario.
/lib/	Es el directorio que contiene todas las bibliotecas esenciales compartidas que usan los programas binarios que están alojados en los directorios bin y sbin. Contiene también las bibliotecas para el núcleo.
/media/	Es el directorio que contiene los puntos de montaje de los medios extraíbles de almacenamiento, tales como lectores de CD-ROM, USB, SSD externas, aunque también se usa para montar otras particiones con el fin de ser utilizadas por otros sistemas.
/mnt/	Es un directorio semejante a media, aunque con un objetivo diferente ya que, en este directorio se suelen montar discos duros y particiones de forma temporal en el sistema. Además, a diferencia del directorio media no requiere de contraseña.
/opt/	Es el directorio que contiene los paquetes de programas opcionales de las aplicaciones que pueden ser compartidas por los usuarios. No así sus configuraciones que se guardan en su respectivo directorio dentro del directorio home.
/proc/	Es el directorio que suele contener los archivos de texto, sistema de archivos virtuales que documentan al núcleo y el estado de los procesos en archivos de texto como puedan ser uptime o network.
/root/	Es el directorio raíz del superusuario o root presenta el mismo funcionamiento que las carpetas que están ubicadas home.

/sbin/	Es el directorio que contiene el sistema de binarios esencial, comandos y programas exclusivos del superusuario como puedan ser init, route o ifup. La ejecución de estos comandos puede ser llevada a cabo por cualquier usuario siempre y cuando tenga los permisos suficientes o tenga la contraseña del superusuario.
/srv/	Es el directorio que contiene los datos y servicios ofrecidos por el sistema tales como los datos de servidores web o servidores FTP. De hecho, cuando un sistema actúa como servidor web, los archivos del sitio web se suelen colocar en la carpeta /srv/www.
/sys/	Es el directorio que contiene el sistema de archivos virtual que almacena y permite la modificación de los dispositivos conectados al sistema.
/tmp/	Es el directorio que contiene los archivos temporales como los archivos del navegador de Internet.
/usr/	Es el directorio que contiene la mayoría de las utilidades y aplicaciones multiusuario accesibles para todos los usuarios en modo de sólo lectura.
/usr/bin/	Es el directorio que contiene la inmensa mayoría de los comandos binarios que usan los usuarios. Aunque son de sólo lectura, pueden tener su propia configuración dentro de la carpeta home.
/usr/include/	Es el directorio que contiene los archivos de cabecera o de inclusión estándar.
/usr/lib/	Es el directorio que contiene las bibliotecas compartidas de los archivos binarios ubicados en /usr/bin/. El objetivo de este directorio es el de ahorrar memoria y proporcionar un mayor orden.
/usr/sbin/	Es el directorio que contiene los archivos binarios no esenciales como demonios o para proporcionar alguna característica que se ejecuta generalmente en ciertas circunstancias, incluyendo el inicio o arranque del sistema.
/usr/share/	Es el directorio que contiene datos compartidos que no dependen de la arquitectura del sistema tales como imágenes, sonidos, plantillas, ...
/usr/src/	Es el directorio que contiene los códigos fuente de algunas aplicaciones. El funcionamiento es idéntico a mnt, con la diferencia de que los usuarios pueden guardar en él los códigos fuente de sus programas y bibliotecas para poder acceder fácilmente, sin problemas de permisos.
/usr/X11Rn/	Es el directorio que contiene la configuración de las X Window System, versión 11, versión n. Este directorio se relaciona con el entorno gráfico.
/usr/local/	Es el directorio que contiene los programas y bibliotecas instaladas localmente y para algunos otros archivos. Habitualmente contiene otros subdirectorios compartidos a modo de sólo lectura específicos del ordenador o servidor que los comparte.
/var/	Es el directorio que contiene archivos variables tales como logs, spool, bases de datos o archivos de correo electrónico temporales. No obstante, también suele usarse como un registro del sistema para encontrar los orígenes de los posibles problemas.
/var/cache/	Es el directorio que contiene la memoria caché de las aplicaciones, aunque también se utiliza el directorio tmp para lo mismo.
/var/crash/	Es el directorio que contiene los datos e información más relevante en lo referente a caídas o errores del sistema operativo.

/var/games/	Es el directorio que contiene los datos variables de los juegos del sistema. Cabe destacar que no es un directorio normalizado ya que muchas veces su objetivo se lleva al directorio home , sin embargo, algunas distribuciones como gnome lo utilizan.
/var/lock/	Es el directorio que contiene los archivos que hacen el seguimiento de los recursos que se están usando en cada momento.
/var/log/	Es el directorio que contiene los archivos de registro.
/var/mail/	Es el directorio que contiene el buzón de correo y/o los mensajes de los usuarios.
/var/opt/	Es el directorio que contiene los datos variables de opt.
/var/run/	Es el directorio que contiene la información acerca del funcionamiento del sistema desde el último arranque o puesta en marcha. Esto es, por ejemplo, los usuarios registrados, los que están online o los demonios que se están ejecutando.
/var/spool/	Es el directorio que contiene cosas como las colas de impresión o el correo no leído.
/var/spool/ mail/	Es el directorio que contiene la ubicación de los correos de usuario desaprobados o eliminados.
/var/tmp/	Es el directorio que contiene los archivos temporales que no se eliminan, aunque se cierre la sesión o se produzca un reinicio.

1.4 ACCESO A FICHEROS

Cuando hablamos de acceso a ficheros o archivos nos referimos a la forma de acceder a los bloques que lo constituyen, independientemente de su organización.

En general, podemos distinguir cuatro formas de acceder a la información:

- **Secuencial**: este modo se caracteriza porque los bloques se leen secuencialmente, uno detrás de otro, de principio a fin. Esto implica que, para localizar o actualizar uno de sus bloques tendremos que ir desde el primero hasta el que proceda para poder manipularlo.
- **Directo**: este modo se caracteriza porque cada bloque puede ser accedido de forma directa con sólo conseguir su identificador relativo de registro o referencia abstracta.
- **Por índice**: este modo se caracteriza porque permite el acceso indirecto a los bloques a través de una clave que es recuperada mediante unas consultas secuenciales a la tabla que contiene la clave, la dirección relativa de cada bloque y acceso directo al registro.
- **Dinámico**: este modo es una forma de expresar que se puede acceder a los archivos a través de cualquiera de los modos anteriormente citados.

1.4.1 Elección del tipo de acceso

Para poder elegir el mejor tipo de acceso es importante tener en cuenta tres factores:

- **Índice de volatilidad**: se dice que un archivo tiene una alta volatilidad cuando se prevé que va a tener un alto porcentaje de adiciones y supresiones debido al añadido o eliminado de registros respecto a la cantidad media de registros que haya en un archivo. Es decir, es la a frecuencia con la que se agregan y/o eliminan registros en un archivo.
- **Índice de actividad**: se dice que un archivo es activo cuando se prevé que va a tener un alto porcentaje de actualización o consulta en un período de tiempo fijo respecto al tiempo medio de registro que se encuentran en el archivo. Es decir, es el porcentaje de registros accedidos o actualizados durante un determinado período de tiempo tomado como promedio.
- **Medida o tamaño:** es la cantidad de información que hay almacenada en un archivo dado.

Aunque el índice de volatilidad y el tamaño puede darnos pistas para una buena elección de acceso, uno de los más claros es el índice de actividad. La razón de ello es que nos proporciona la información de si un archivo puede o debe utilizar una organización secuencial o relativa.

1.5 DERECHOS DE ACCESO

Los derechos de acceso son una parte fundamental en la seguridad de publicación de páginas web pues permiten controlar que los archivos y directorios puedan ser manipulados de forma controlada y segura por los usuarios y grupos.

Esta manipulación no sólo se refiere al acceso, sino que también hace referencia a la escritura, la ejecución, al tipo de usuario que es y si es acceso público o privado.

1.5.1 Permisos de acceso para GNU/Linux

Los niveles de permisos de usuario, también denominados tipos de permisos en Linux es un concepto que hace referencia a distintas circunstancias o características y se dividen en tres grupos denominados de propietario, de grupo y de resto de usuarios.

- **Permisos de propietario**: el permiso de propietario es aquel que se asigna cuando se crea un archivo o directorio, normalmente dentro de su directorio de trabajo (habitualmente denominado como su home). Este usuario será quien inicialmente tendrá acceso a toda la información y la posibilidad de hacer lo que desee con ella. Usualmente se le identifica en el sistema con el parámetro "u".

- **Permisos del grupo**: el permiso de grupo es aquel que afecta a todos los usuarios que conforman un grupo. Este tipo de permisos es hereditario, es decir, al momento de incorporar un usuario a un grupo, hereda o se le asignan de forma automática los permisos que tenga todo el grupo. Usualmente se le identifica en el sistema con el parámetro "g".

- **Permisos del resto de usuarios**: el permiso de resto de usuarios es la forma de hacer referencia a los usuarios que no pertenecen al grupo de trabajo donde se encuentra el archivo o directorio. Son unos derechos de acceso que son establecidos por el propietario y, usualmente, se le identifica en el sistema con el parámetro "o".

Para conocer los permisos y otros datos del fichero o directorio se podrá ejecutar el comando **ls -l** de Linux, el cual mostrará similar a lo siguiente:

```
# ls -l
total 16
drwxr-xr-x    2 root     root          2091 Jan  9  2021 js
drwxr-xr-x    4 root     root          2242 Apr 17 16:03 css
drwxr-xr-x   39 root     root          1986 Jan  9  2021 img
-rw-r--r--    1 root     root           151 Jul  5  2020 index.html
```

Viendo el código anterior podemos ver que el archivo **index.html** pertenece al usuario root, también denominado superusuario, pero que podrá ser leído por otros usuarios o grupo de ellos.

Si deseamos un nivel más detallado del contenido del directorio donde nos encontramos podremos hacer algo como:

```
# ls -alt
total 68
drwxrwxrwx   21 root     root           461 Apr 17 16:13 .
drwxrwxrwx   21 root     root           461 Apr 17 16:13 ..
drwxr-xr-x    2 root     root          2091 Jan  9  2021 js
drwxr-xr-x    4 root     root          2242 Apr 17 16:03 css
drwxr-xr-x   39 root     root          1986 Jan  9  2021 img
-rwxr-xr-x   39 root     root          1986 Jan  9  2021 .htaccess
-rw-r--r--    1 root     root           151 Jul  5  2020 index.html
```

En un principio, el superusuario o administrador será el encargado de dar de alta a los usuarios y asignarles, como mínimo, un grupo. No obstante, será frecuente encontrar que un usuario esté en más de un grupo.

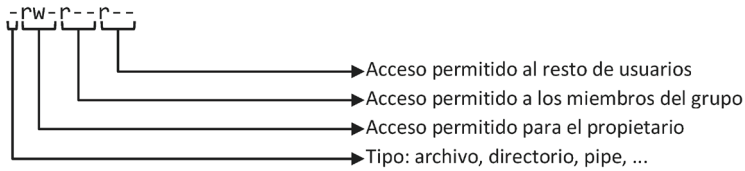
Asimismo, cada usuario o grupo podrá tener permisos de lectura, escritura y ejecución globales o independientes. Esto es, podrán tener permisos que afecten a uno o varios usuarios y uno o varios grupos que, a su vez, podrán tener únicamente asignados los permisos de ejecutar, leer y/o escribir.

En general, el propietario del archivo será el usuario que lo creó, pero podrá darse el caso de que cambie de dueño. No obstante, dentro de este contexto, lo importante es que, en general, el propietario del archivo es el que concede permisos a los usuarios, sean o no de su mismo grupo.

Una vez dicho esto, veamos qué significan los 10 caracteres de la izquierda del listado que acabamos de ver. Sin embargo, por intentar ser algo más gráfico, escojamos como ejemplo el archivo **index.html**.

```
-rw-r--r--    1 root     root           151 Jul  5  2020 index.html
```

Al contemplar la combinación de caracteres podremos extraer cuatro zonas:



Como se puede observar, el primer carácter hace referencia al tipo. Esto es, nos indicará que tipo de "archivo" es.

Contenido	Significado
-	Archivo (común).
d	Directorio.
c	Dispositivo de caracteres (TTY o impresora).
b	Dispositivo de bloque (usualmente disco duro o CD-ROM).
l	Enlace simbólico.
s	Socket.
p	Pipe.

En lo referente a los siguientes caracteres, las columnas 2, 3 y 4 valdrán para saber qué se aplica al propietario, las columnas 5, 6 y 7 valdrán para saber qué se aplica al grupo y las columnas 8, 9 y 10 valdrán para saber qué se aplica al resto.

Como apunte final, recordemos que los permisos son hereditarios y, por esta razón, podríamos encontrarnos con una situación donde un archivo tiene todos los permisos (lectura, escritura y ejecución), pero no es posible acceder a él. Esto se debe, en su inmensa mayoría de casos, a que el directorio donde está ubicado el archivo al que queremos acceder no cuenta con los permisos adecuados.

1.5.1.1 CAMBIANDO LOS PERMISOS EN LINUX

Para cambiar los derechos de acceso de un archivo o directorio en Linux se puede recurrir al comando **chmod**, que viene de la abreviatura de "change mode", y cuya forma genérica podría definirse como:

```
# chmod {u, g, o, a}{+, -, =}{r, w, x} nombre
```

Por ejemplo, si queremos eliminar el permiso de escritura de nuestro archivo **index.html** para nuestro grupo, podríamos ejecutar el siguiente comando:

```
# chmod g-w index.html
```

Sin embargo, como los permisos pueden tratarse como columnas o bits, también podremos hacer cambios usando su valor en binario. Esto es, si entendemos que el guion equivale a 0 y que la combinación rwx equivale a 111 (un valor de 1 por cada letra), podremos cambiar los derechos de acceso a través de la siguiente forma:

```
# chmod 777 index.html
# chmod 000 index.html
# chmod 644 index.html
```

Así, por ejemplo, el valor de rwx es 7, el valor de rw- es 6, el valor de r-- es 4 y el valor de --- es 0.

1.5.1.2 CAMBIANDO GRUPOS Y USUARIOS EN LINUX

Para cambiar el propietario de uno o varios archivos y directorios se puede recurrir al comando **chown**, que viene de la abreviatura "change own", y cuya forma genérica podría definirse como:

```
# chown [opciones] usuario[:grupo] nombre(s)
```

En lo referente a las opciones tenemos las siguientes:

Contenido	Significado
-R	Cambia el propietario y el grupo (opcionalmente) para todos sus contenidos de forma recursiva. # chown -R root /var/www/html # chown -R user1:htusers /var/www/html
-v (o --verbose)	Muestra una respuesta o salida más descriptiva que proporciona un diagnóstico para cada entrada procesada, aunque no se haya producido ningún cambio. # chown -R --verbose root /var/www/html
-c (o --changes)	Muestra una respuesta o salida únicamente cuando se haya realizado algún cambio. # chown -R -c root /var/www/html
--dereference	Hace que actúe sobre los enlaces simbólicos en vez de hacerlo sobre el destino. # chown -dereference user symbollink
-h (o --no-dereference)	Cambia el propietario del destino en lugar del propio enlace, siempre y cuando se trate de enlaces simbólicos. # chown -h user symbollink
--reference	Asigna como propietario de un archivo al propietario del archivo pasado en la referencia. # chown --reference=index.html index2.html
-f (--silent , --quiet)	Provoca que no se muestren los errores. # chown -fR root /var/www/html
--preserve-root	Evita que actúe de forma recursiva en /. Esta opción debe usarse con **-R** opción para que surta efecto. # chown -cfR --preserve-root user1 /

En lo referente al usuario se puede utilizar su nombre o el número de identidad de usuario (UID). Para el grupo se debe añadir el signo de dos puntos seguido del nombre que se desee asignar.

1.5.1.3 UTILIDAD UMASK

La utilidad **umask**, abreviatura de "User Mask" y más conocida como "Máscara de Usuario" o "Máscara de creación de archivos de usuario" es un permiso base que se establece por defecto cuando se crea un nuevo archivo o carpeta.

Es utilizado por **mkdir**, **touch**, **tee**, entre otros comandos y sirve para establecer los permisos por defecto cuando se crean archivos o directorios, Esta máscara de usuario interviene en todos y cada uno de los pasos que se dan durante el proceso de creación de un nuevo archivo o directorio y suele ser el administrador o superusuario el que la establece durante el proceso de instalación de Linux.

La forma genérica podríamos definirla como:

```
# umask [-S] [máscara]
```

Si no ponemos ninguno de los parámetros se nos mostrará la máscara que tiene el sistema establecido por defecto.

```
# umask
0022
```

Esto significa que, cada vez que creemos un archivo o directorio, se le restarán estos valores a los valores por defecto del sistema que son, 666 para los archivos y 777 para los directorios.

Para entender esto mejor veamos un ejemplo. Para ello, lo primero que haremos es crear un archivo llamado **prueba.html** y un directorio **dirAux**:

```
# touch prueba.html
# mkdir dirAux
```

Al realizar estas acciones y realizar un **ls -l** deberíamos ver algo similar a lo siguiente:

```
Output:
-rw-r--r--   2 root     root          4096 Apr  14  20:23 prueba.html
drwxr-xr-x   2 root     root             0 Apr  14  20:23 dirAux
```

Si nos fijamos en la ilustración anterior nos daremos cuenta de que los permisos que se han asignado son el resultado de restar su valor por defecto con la máscara de usuario. Esto es, para nuestro archivo **prueba.html** 644 y, para nuestro directorio **dirAux** 755.

```
prueba.html         dirAux
  0666                0777
- 0022              - 0022
  ----                ----
  0644                0755
```

Por tanto, los permisos resultantes para el archivo que el propietario tiene permisos de lectura y escritura y, tanto el grupo como el resto, sólo tienen permiso de lectura.

Para el directorio pasa un poco lo mismo. Los permisos resultantes que el propietario tiene permisos de lectura, escritura y ejecución y, tanto el grupo como el resto, tienen permiso de lectura y ejecución.

A continuación, se muestran los valores más habituales que tiene **umask**:

Valor	Acceso del Usuario	Accesos del grupo	Accesos para el resto
0000	Todos	Todos	Todos
0002	Todos	Todos	Lectura, ejecución
0007	Todos	Todos	Ninguno
0022	Todos	Lectura, ejecución	Lectura, ejecución
0027	Todos	Lectura, ejecución	Ninguno
0077	Todos	Ninguno	Ninguno

Hasta ahora hemos visto la forma de trabajar con **umask**, sin embargo, nos falta comentar cómo ver los permisos en formato simbólico, lo que mostrábamos en la forma genérica con **-S**:

```
# umask -S
u=rwx,g=rx,o=rx
```

Si ejecutamos la instrucción anterior, lo más probable es que nos muestre el valor indicado, pues se corresponde con el valor de máscara 0022.

ⓘ NOTA FINAL

En la mayoría de las distribuciones, el valor por defecto para todo el sistema está establecido en el archivo **pam_umask.so** o en **/etc/profile**. Sin embargo, existe la posibilidad de personalizarlo para cada usuario añadiendo el valor deseado en el archivo **~/.bashrc** del directorio personal.

1.5.2 Permisos de acceso para Windows

Los niveles de permisos de usuario en Windows, comúnmente denominados tipos de cuenta, son administrador y usuario, sin embargo, dentro del nivel de usuario podemos encontrar dos subtipos en función del uso que se va a dar.

La principal diferencia entre un tipo de cuenta y otra son los permisos a los que se tendrá acceso sobre las diferentes funciones del sistema, lo que nos hará tener que pensar bien qué necesidades va a tener el usuario. Sin embargo, aunque en un principio le asignemos un tipo de cuenta concreta, posteriormente podremos cambiársela si así las necesidades lo requiriesen.

1.5.2.1 TIPOS DE CUENTA

Cuenta de usuario con permisos estándar

En este caso, únicamente podremos realizar acciones funcionales en el sistema, como por ejemplo cambiar los iconos, el fondo de pantalla o cualquier otra acción que esté asociada al "trabajo diario".

En general, no podremos editar o actualizar el registro de Windows, instalar o desinstalar aplicaciones que afecten directamente al sistema ni tener acceso a determinadas partes de la configuración del entorno y sistema.

Su aplicación más frecuente suele darse en usuarios que únicamente requieren manipular documentos y archivos, acceder a Internet y a redes sociales, reproducir vídeos y otras acciones no demasiado invasivas en lo que al sistema y equipo se refiere.

Cuenta de administrador

Si se desea ir más allá necesitaremos tener permisos de administrador. Para ello deberemos loguearnos como Administrador pulsando en el botón o menú de **Inicio** / **Icono de usuario actual** / **Cambiar de cuenta** o introducir la contraseña del administrador cuando se desee realizar una acción que no permite la cuenta de usuario estándar.

Por tanto, como ya se habrá podido deducir, el usuario administrador será el indicado para realizar cualquier cambio en el sistema y de configuración general o de permisos de usuario.

Su aplicación más frecuente suele darse en usuarios que requieren actualizar el registro de Windows, instalar o desinstalar aplicaciones de sistema o que afectan al mismo y otro tipo de acciones que no permite la cuenta de usuario estándar.

Cuenta de invitado

Por último, existe un tipo de cuenta denominada "invitado" que está pensada para aquellos usuarios que sólo van a hacer un uso liviano del sistema. No permite el acceso a los archivos personales de otros usuarios y tampoco permite hacer cambios en la configuración del sistema, lo que hace que sea una buena opción si lo que se desea es proteger la privacidad y seguridad de los demás usuarios dados de alta.

1.5.2.2 TIPOS DE PERMISOS EN WINDOWS

En Windows podemos encontrar cinco tipos de permisos básicos que definen o permiten un conjunto o serie de acciones diferentes que se pueden realizar. En concreto, estos permisos son **lectura**, **lectura y ejecución**, **modificación**, **escritura** y **control total**.

No obstante, para las carpetas también podemos personalizar los permisos o definir variaciones de los niveles de permisos estándar ya que, dentro de cada uno de los niveles de permiso se pueden dar múltiples posibilidades. A continuación, se muestran las limitaciones de cada uno de ellos:

Permiso	Significado
Control Total	Mostrar el nombre del archivo y directorios. Navegar por los subdirectorios. Mostrar datos en los archivos del directorio. Agregar archivos y subdirectorios al directorio. Modificar los archivos del directorio. Eliminar el directorio con sus archivos. Cambiar permisos. Volvernos propietarios del directorio y sus archivos.
Modificar	Mostrar el nombre del archivo y directorios. Navegar por los subdirectorios. Mostrar datos en los archivos del directorio. Agregar archivos y subdirectorios al directorio. Modificar los archivos del directorio. Eliminar el directorio con sus archivos. Abrir y modificar archivos.
Leer	Mostrar el nombre del archivo y directorios. Navegar por los subdirectorios. Abrir archivos. Copiar y ver datos en los archivos del directorio.
Leer y ejecutar	Mostrar el nombre del archivo y directorios. Navegar por los subdirectorios. Mostrar datos en los archivos del directorio. Ejecutar aplicaciones.
Escribir	Añadir nuevos archivos. Crear carpetas. Eliminar Archivos.

1.5.2.3 PERMISOS NTFS DE WINDOWS

Cuando hablamos de permisos NTFS lo primero que se debe diferenciar es si estamos hablando de archivos o de directorios. En lo referente a ficheros, es posible proporcionar permisos de **lectura**, **escritura** y **edición o modificación**. Sin embargo, en lo referente a carpetas o directorios, cuando se posee **control total** sobre una carpeta, además se hace posible **borrar** ficheros y subdirectorios de la carpeta principal.

A continuación, se muestran las restricciones de acceso para cada permiso NTFS.

Permiso	Control Total	Modificar	Leer y ejecutar	Mostrar contenido de la carpeta (sólo carpetas)	Lectura	Escritura
Atributos de escritura	X	X				X
Atributos de escritura extendidos	X	X				X
Atributos de lectura	X	X	X	X	X	
Atributos de lectura extendidos	X	X	X	X	X	
Cambiar permisos	X					
Crear archivos / Escribir datos	X	X				X
Crear carpetas / Anexar datos	X	X				X
Eliminar	X	X				
Eliminar subcarpetas y archivos	X					
Leer permisos	X	X	X	X	X	X
Listar carpeta / Leer datos	X	X	X	X	X	
Recorrer carpeta / Ejecutar archivo	X	X	X	X		
Sincronizar	X	X	X	X	X	X
Tomar posesión	X					

Atributos de lectura y escritura

Son aquellos que permiten o impiden que el usuario cambie los atributos definidos por el sistema sobre un archivo o carpeta. Estos son, por ejemplo, los atributos de sólo lectura y oculto.

No obstante, cabe destacar que cuando es de escritura su asignación no implica la creación o eliminación de archivos o directorios, únicamente el poder hacer cambios en los atributos de un archivo o de una carpeta.

Atributos de lectura y escritura extendidos

Son aquellos que permiten o impiden que el usuario vea los atributos que no son del propio sistema, es decir, aquellos que están definidos por los programas o aplicaciones que son dependientes de ellas.

Al igual que sucede con los permisos de lectura y escritura, su asignación no implica la creación o eliminación de archivos o directorios, únicamente el poder hacer cambios en los atributos de un archivo o de una carpeta.

Cambiar permisos

Consiente o impide que el usuario cambie los permisos del archivo o directorio como, por ejemplo, **Control total**, **Leer** y/o **Escribir**.

Crear archivos / Escribir datos

Mientras que el permiso de **crear archivos** sólo es aplicable a carpetas y consiente o impide que el usuario cree archivos en el directorio, el permiso **escribir datos** sólo es aplicable a archivos y consiente o impide que el usuario edite el archivo y sobrescriba su contenido.

Crear carpetas / Anexar datos

Mientras que el permiso de **crear carpetas** sólo es aplicable a carpetas y consiente o impide que el usuario cree subdirectorios en el directorio, el permiso **anexar datos** sólo es aplicable a archivos y consiente o impide que el usuario haga cambios al final del archivo, pero sin que cambie, elimine ni sobrescriba los datos existentes.

Eliminar

Consiente o impide que el usuario elimine el archivo o la carpeta. No obstante, si no dispone de este permiso para un archivo o directorio dado, igualmente podrá eliminarlo si se posee permiso de **Eliminar subcarpetas y archivos** en la carpeta principal.

Eliminar subcarpetas y archivos

Consiente o impide que el usuario elimine subcarpetas y/o archivos, incluso si no posee el permiso **Eliminar** para la subcarpeta o el archivo. Además, sólo es aplicable a directorios.

Leer permisos

Consiente o impide que el usuario visualice o lea los permisos del archivo o directorio como, por ejemplo, **Control total**, **Leer** y/o **Escribir**.

Listar carpeta / Leer datos

Mientras que el permiso de **listar carpeta** sólo es aplicable al contenido de los directorios y consiente o impide que el usuario vea los nombres de archivo y de subcarpeta de la carpeta, el permiso **leer datos** sólo es aplicable a archivos y consiente o impide que el usuario vea los datos de los archivos.

Recorrer carpeta / Ejecutar archivo

Mientras que el permiso de **recorrer carpeta** sólo es aplicable a directorios y consiente o impide que el usuario acceda de una carpeta a otra, incluso si el usuario no posee los permisos para las carpetas recorridas, el permiso de **ejecutar archivos** sólo es aplicable a archivos y consiente o impide los archivos de programa o aplicación que hay en ejecución.

En este contexto cabe destacar dos cosas:

- El permiso de **recorrer carpeta** sólo proporcionará el resultado esperado siempre y cuando el usuario o grupo no tenga el permiso **omitir comprobación de recorrido** activado, el cual comprueba los derechos de acceso en la directiva de grupo.

- La configuración del permiso **recorrer carpeta** en un directorio no tiene por qué establecer automáticamente el permiso **ejecutar archivo** en todos los archivos de dicho directorio.

Sincronizar

Este permiso sólo es aplicable a aplicaciones o programas multiproceso con varios subprocesos y consiente o impide que dichos subprocesos esperen en el controlador del archivo o de la carpeta para sincronizarse con otro subproceso.

Tomar posesión

Este permiso consiente o impide que el usuario tome posesión del archivo o directorio. En general, se puede afirmar que el propietario de un archivo o un directorio dado puede cambiar sus permisos, sean cuales sean los permisos existentes que lo protegen.

1.5.2.4 ADQUIRIENDO PERMISOS DE ADMINISTRADOR

A través de la interfaz gráfica

Para cambiar los derechos de acceso a través de la interfaz gráfica podemos recurrir a la página de Configuración o al Panel de Control.

Si elegimos la vía a través de la pantalla de Configuración deberemos pulsar en el botón o menú de **Inicio** / **Configuración** / **Cuentas** / **Familia y otros usuarios**. Si elegimos la vía a través del Panel de Control deberemos pulsar en el botón o menú de **Inicio** / **Panel de Control** / **Cuentas de Usuario**.

Una vez que estemos dentro de esta pantalla se nos mostrará un listado con todos los usuarios del sistema en donde podremos seleccionar el que nos interesa y, a continuación, pulsar en el botón **Cambiar tipo de cuenta**.

Una vez que hayamos realizado el cambio pertinente, para que se vuelva efectivo deberemos cerrar la sesión y volver a logarnos.

A través de la línea de comandos

Para cambiar permisos de administrador a través de la línea de comandos sólo deberemos abrir el **símbolo de sistema (cmd)** y ejecutar:

```
C:\>Net Localgroup Administrators elnombredeusuarioquequieres /add
```

Si el proceso se completa con éxito y se nos muestra un mensaje de confirmación, al igual que antes, bastará con cerrar la sesión y volver a logarnos para que los cambios se hagan efectivos.

1.5.2.5 CAMBIANDO LOS PERMISOS EN WINDOWS

Para cambiar los derechos de acceso de un archivo o directorio en Windows se puede recurrir a la interfaz gráfica pulsando en el botón **derecho del ratón**, seleccionando la opción de **Propiedades** y pulsando en la pestaña de **Seguridad**, sin embargo, nosotros aquí veremos cómo hacerlo a través de símbolo de sistema y el comando **icacls**, que viene de la abreviatura de "Integrity Control Access Control List", y cuya forma genérica podría definirse como:

```
C:\>icacls [nombre] [opciones usuario:máscara] [flags] [nivel_integridad]
```

El nombre hace referencia al nombre del fichero o directorio y las opciones habitualmente son palabras clave que indican la acción que se va a llevar a cabo.

En lo referente al usuario y la máscara servirán para indicar el usuario y permisos que se desean aplicar. La máscara puede especificarse de dos maneras:

A través de la secuencia de derechos simples:

Código	Significado
N	Sin permiso.
F	Control total.
M	Permiso de modificación.
RX	Permiso de lectura y ejecución.
R	Permiso de solo lectura.
W	Permiso de solo escritura.
D	Permiso de borrado o eliminación.

A través de una lista separada por comas entre paréntesis:

Código	Significado
DE	Eliminar.
RC	Control de lectura.
WDAC	Escribir DAC.
WO	Escribir propietario.
S	Sincronizar.
AS	Acceso al sistema de seguridad.
MA	Máximo permitido.
GR	Lectura genérica.
GW	Escritura genérica.
GE	Ejecución genérica.
GA	Todo genérico.
RD	Leer datos/lista de directorio.
WD	Escribir datos/agregar archivo.
AD	Anexar datos/agregar subdirectorio.
REA	Leer atributos extendidos.
WEA	Escribir atributos extendidos.
X	Ejecutar/atravesar.
DC	Eliminar secundario.
RA	Leer atributos.
WA	Escribir atributos.
OI	Herencia de objeto. Puede preceder a cualquier forma y se aplica sólo a directorios.
CI	Herencia de contenedor. Puede preceder a cualquier forma y se aplica sólo a directorios.
IO	Sólo herencia. Puede preceder a cualquier forma y se aplica sólo a directorios.
NP	No propagar herencia. Puede preceder a cualquier forma y se aplica sólo a directorios.
I	Permiso heredado del contenedor principal. Puede preceder a cualquier forma y se aplica sólo a directorios.

Entre las opciones más comunes tenemos:

Ópción	Significado
/T	Indica que esta operación se realiza en todos los archivos o directorios coincidentes bajo los directorios especificados en el nombre.
/C	Indica que esta operación continuará en todos los errores de archivo. Se seguirán mostrando los mensajes de error.
/L	Indica que esta operación se realiza en el vínculo simbólico en sí en lugar de en su destino.
/Q	Indica que icacls debe suprimir los mensajes de que las operaciones se realizaron correctamente.

Para más información se puede visitar la siguiente dirección:

Comando icacls	Código QR
Muestra o modifica las listas de control de acceso discrecional (DACL) en los archivos especificados y aplica las DACL almacenadas a los archivos de los directorios especificados. *https://learn.microsoft.com/es-es/windows-server/administration/windows-commands/icacls*	

Dicho esto, ahora pasaremos a poner unos pocos ejemplos para que su entendimiento sea algo más clarificador.

Por ejemplo, si queremos enumerar los permisos NTFS actuales en la carpeta C:\CURSOS podremos hacer algo como:

```
C:\>icacls "C:\CURSOS"
cursos NT AUTHORITY\Usuarios autentificados:(I)(M)
       NT AUTHORITY\Usuarios autentificados:(I)(OI)(CI)(IO)(M)
       NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
       BUILTIN\Administradores:(I)(OI)(CI)(F)
       BUILTIN\Usuarios:(I)(OI)(CI)(RX)

Se procesaron correctamente 1 archivos; error al procesar 0 archivos
```

Pero, si queremos proporcionar permisos de modificación para el contenido de la carpeta cursos al usuario webuser podremos hacer algo como:

```
C:\>icacls "C:\CURSOS" /grant webuser:M
```

ⓘ NOTA

Cabe destacar que puede ocurrir que la ejecución de estos comandos haga que se muestre el mensaje de "Acceso denegado". Esto sucede al intentar cambiar los permisos en un archivo o directorio y no tener permisos para hacerlo. Si se da este caso lo que deberemos comprobar antes de nada es que **cmd** se está ejecutando como administrador.

Ahora bien, si lo que deseamos es proporcionar permiso de **control total** al directorio C:\CURSOS sobre el grupo web y aplicar todas las configuraciones a sus subdirectorios podremos hacer algo como:

```
C:\>icacls "C:\CURSOS" /grant webuser:F /Q /C /T
```

No obstante, hay otras formas de asignar permisos. Por ejemplo, para proporcionar al usuario webuser permisos de lectura, ejecución y eliminación sobre la carpeta C:\CURSOS podríamos hacer algo como:

```
C:\>icacls "C:\CURSOS" /grant webuser:(OI)(CI)(RX,D)
```

Y, si deseamos eliminar todos los permisos NTFS asignados al mismo usuario podríamos hacer:

```
C:\> icacls "C:\CURSOS" /remove webuser
```

Otra cosa común que podríamos necesitar es denegar el acceso a los archivos de C:\CURSOS al usuario webuser:

```
C:\>icacls "C:\CURSOS" /deny " webuser:(CI)(M)"
```

¿Y cómo hacemos para habilitar los permisos heredados en la carpeta C:\CURSOS? Pues la respuesta debería ser algo como:

```
C:\>icacls "C:\CURSOS" /inheritance:e
```

¿Y para deshabilitarlos? Pues podríamos hacer algo como:

```
C:\>icacls "C:\CURSOS" /inheritance:r
```

Y, por último, pero no menos importante, podríamos necesitar cambiar el propietario. Para ello, supongamos que deseamos cambiar el propietario del archivo C:\CURSOS\Curso1.docx a webuser. El comando sería algo como:

```
C:\>icacls "C:\CURSOS\Curso1.docx" /setowner webuser
```

Y, si deseamos cambiar el propietario para todos los archivos del directorio C:\CURSOS, ¿cómo lo podemos hacer? La respuesta está a continuación:

```
C:\>icacls "C:\CURSOS\*" /setowner webuser /T /C /L /Q
```

1.6 ÓRDENES DE CREACIÓN, MODIFICACIÓN Y BORRADO

La creación, modificación y eliminación de archivos y directorios se realiza de forma bastante similar cuando se está trabajando bajo una interfaz gráfica, sin embargo, la cosa cambia mucho cuando estamos en línea de comandos.

Por ejemplo, mientras que para ver el directorio actual en Linux se recurre al comando **pwd**, en Windows se consigue a través del comando **cd**. A continuación, se muestran los diferentes comandos para cada sistema:

1.6.1 Bajo el sistema operativo GNU/Linux

Acción	Comando	Comentarios
Mostrar directorio actual	pwd	Muestra el nombre del directorio actual.
Cambiar de directorio	cd [nombre]	Si no se especifica un nombre, el comando solicitará el cambio al directorio personal del usuario, que en Linux suele ser **/home**.
Crear directorio	mkdir nombre1 [nombre2...]	Los directorios deben estar separados por espacios.
Eliminar directorio	rmdir nombre1 [nombre2...]	Elimina el directorio o directorios especificado(s) siempre y cuando esté(n) vacío(s) y no sea el actual.
Renombrar directorio	mv [nombre] [nombre_nuevo]	El nombre y nombre_nuevo deben estar separados por un espacio.
Mostrar contenido	cat [-n] nombre1 [nombre2...]	Muestra por pantalla el contenido de los archivos que se le pasen en la lista concatenados uno detrás de otro. Si establecemos **-n** se mostrarán las líneas enumeradas.
Crear archivo o vaciar existente	touch nombre1 [nombre2...]	En el caso de que no exista el nombre indicado el archivo se creará. En caso contrario, se eliminará todo el contenido del archivo dejándolo totalmente vacío.
Editar archivo	nano [-v][-w] [nombre]	En caso de no indicar un nombre se pedirá un nombre al realizar el guardado. Si establecemos **-v** el archivo se abrirá en modo de sólo lectura. Para abrir un archivo de configuración del sistema se requerirá la opción **-w**.
Editar archivo	vim [nombre]	**Vim** es un editor de texto con una mayor complejidad de uso, pero iguales resultados que **nano**. Al igual que **nano**, en caso de no indicar un nombre se pedirá un nombre al realizar el guardado.
Copiar archivos	cp [-r] origen destino	Tanto el origen como el destino pueden ser rutas relativas o absolutas. Si establecemos **-r** copiará recursivamente todos sus subdirectorios y archivos internos.
Mover archivos	mv nombre nombrenuevo	El nombre y nombrenuevo deben estar separados por un espacio.
Eliminar archivos	rm [-r] nombre1 [nombre2...]	Los nombres deben estar separados por un espacio. Si establecemos **-r** y el elemento es un directorio, se eliminarán recursivamente todos sus subdirectorios y archivos internos.
Enlaces simbóllicos y accesos directos	ln nombre nombre_ nuevo	Establece un vínculo entre dos ficheros, de manera que al actualizar uno de ellos se actualice el otro. El comando **ln** presenta múltiples opciones que influyen en la eliminación y/o creación de los enlaces simbólicos.

1.6.2 Bajo el sistema operativo Windows

Acción	Comando	Comentarios
Mostrar directorio actual	cd chdir	Muestra el nombre del directorio actual.
Cambiar de directorio	cd nombre chdir nombre	Si no se especifica un nombre, el comando solicitará el cambio al directorio personal del usuario, que en Linux suele ser **/home**.
Crear directorio	md nombre1 [nombre2...] mkdir nombre1 [nombre2...]	Los directorios deben estar separados por espacios.
Eliminar directorio	rd nombre1 [nombre2...] rmdir nombre1 [nombre2...]	Elimina el directorio o directorios especificado(s) siempre y cuando esté(n) vacío(s) y no sea el actual.
Renombrar directorio	move nombre nombrenuevo	El nombre y nombrenuevo deben estar separados por un espacio.
Copiar directorio	copy origen destino	Tanto el origen como el destino pueden ser rutas relativas o absolutas.
Ver el arbol de contenidos	tree nombre	Si no se especifica un directorio se mostrará el árbol de directorios desde el directorio actual.
Mostrar contenido	type nombre1 [nombre2...]	Muestra por pantalla el contenido de los archivos que se le pasen en la lista concatenados uno detrás de otro.
Crear archivo	fsutil file createnew nombre	Crea un archivo con el nombre especificado. El nombre puede llevar una ruta relativa o absoluta.
Editar archivo	edit nombre	Inicia el Editor de MS-DOS, que crea y cambia los archivos de texto ASCII. Si esta utilidad no está disponible en Windows se puede recurrir a cambiar **edit** por **notepad**.
Copiar archivos	copy origen destino	Tanto el origen como el destino pueden ser rutas relativas o absolutas. Aunque aquí no se muestran, presenta múltiples opciones para diferentes situaciones como la supresión de confirmación en la sobreescritura.
Mover archivos	move nombre nombrenuevo rename nombre nombrenuevo	El nombre y nombrenuevo deben estar separados por un espacio.
Eliminar archivos	del nombre1 [nombre2...] erase nombre1 [nombre2...]	Los nombres deben estar separados por un espacio. Aunque aquí no se muestran, presenta múltiples opciones para diferentes situaciones como no pedir confirmación o forzar su eliminación, aunque esté en modo de sólo lectura.