

UNIDAD 3: GESTIÓN DE LA INFORMACIÓN

Módulo Profesional: Sistemas Informáticos

Índice

RESUMEN INTRODUCTORIO	4
INTRODUCCIÓN	4
CASO INTRODUCTORIO.....	5
1. ESTRUCTURA DE ALMACENAMIENTO DE LA INFORMACIÓN. SISTEMAS DE ARCHIVOS.	6
1.1 Sistemas de archivos Linux.....	9
1.2 Sistemas de archivos Windows.....	9
1.3. Sistemas de archivos en red	10
1.4. Sistemas de archivos Macintosh.	10
2. ORGANIZACIÓN DE ARCHIVOS.....	11
2.1. Sistema binario	11
2.1.1. Conversión del sistema binario al decimal.	11
2.1.2. Conversión del sistema decimal a binario	12
2.2. Unidades de almacenamiento de datos.....	13
2.3 Jerarquía de datos	13
2.4. Características de los archivos	14
2.5. Tipos de archivos	17
3. FICHEROS Y DIRECTORIOS	18
3.1 Operaciones con ficheros.....	18
3.2 Operaciones con directorios	19
3.3 Estructura de directorios en Linux.....	21
4. GESTIÓN DE ARCHIVOS MEDIANTE COMANDOS Y ENTORNOS GRÁFICOS	24
4.1 Sistemas Windows.....	24
4.1.1 Utilización de comandos. Símbolo del sistema	24
4.1.2 Entorno gráfico. Explorador de Windows	27
4.2 Sistemas Linux.....	30
4.2.1 Utilización de comandos. Terminal	30
4.2.2 Entorno gráfico. Archivos	33
5. VARIABLES DE ENTORNO.....	36
5.1 El PATH.....	37
5.2 Variables de entorno en sistemas Windows.....	39

5.3 Variables de entorno en sistemas Linux.....	42
6. ADMINISTRACIÓN DE DISCOS.....	45
6.1 Particiones, unidades lógicas y volúmenes.....	46
6.1.1 Creación de particiones en Windows 10	48
6.1.2. Particiones en Linux	53
6.2 Desfragmentación	54
6.3. Copias de seguridad	55
6.3.1 Tipos de copias de seguridad.....	55
6.3.2 Creación de copias de seguridad	58
6.4. RAID.....	60
7. AUTOMATIZACIÓN DE TAREAS	65
RESUMEN FINAL	70

RESUMEN INTRODUCTORIO

A lo largo de esta unidad conoceremos las principales técnicas de organización de la información en sistemas informáticos, revisando el concepto de sistema de archivos y su utilización en los distintos sistemas operativos.

Estudiaremos también las operaciones más habituales con archivos y directorios y veremos el modo de llevarlas a cabo, tanto en modo gráfico como mediante comandos, en sistemas operativos libres y propietarios.

Trabajaremos con las variables de entorno tanto en sistemas Windows como Linux, haciendo hincapié en la facilidad que proporcionan a la hora de acceder a los programas instalados.

Finalmente revisaremos las herramientas de administración de discos, que nos permitirán hacer un mejor uso del espacio de almacenamiento disponible, tanto en lo que se refiere a organizar nuestros datos como a proporcionar mayor seguridad en el acceso a ellos.

INTRODUCCIÓN

La gestión de la información es uno de los servicios más destacables de cualquier sistema operativo. Los computadores pueden almacenar información de diversas formas y cada dispositivo de almacenamiento tiene sus propias características y organización física. Como desarrolladores es fundamental conocer perfectamente cada una de estas formas, ya que las aplicaciones que generemos deberán funcionar correctamente en cualquier sistema.

El sistema operativo ofrece una perspectiva lógica y uniforme del almacenamiento de la información y abstrae las propiedades físicas de sus dispositivos de almacenamiento para definir una unidad lógica de almacenamiento, válida para todos los dispositivos: el archivo. Cómo organizar dichos archivos, estableciendo una estructura jerárquica mediante carpetas, será una de las tareas básicas para cualquiera que desee utilizar un sistema informático. Eso nos permitirá disponer de los datos necesarios en cada momento sin tener que realizar búsquedas de forma continuada, lo que redundará en un trabajo más rápido y efectivo.

Y, por supuesto, ser capaz de administrar nuestros sistemas de almacenamiento de forma óptima, consiguiendo que resulten fiables y seguros, es otra competencia fundamental para un desarrollador.

CASO INTRODUCTORIO

Comienzas a trabajar como desarrollador web en una pequeña empresa, y te das cuenta de que la gestión de la información que realizan actualmente no es la más óptima. Sus datos no están bien estructurados, no hay particiones en los discos de sus equipos, no se realizan copias de seguridad de forma planificada...

Te ofreces a mejorar sus sistemas de almacenamiento, planteando las ventajas que ello reportará a todos los trabajadores de la empresa.

Al finalizar la unidad el alumnado conocerá la importancia de estructurar el almacenamiento de información en un sistema informático, identificará los principales sistemas de archivos y sus características, será capaz de realizar las principales tareas de gestión de archivos y directorios, tanto de modo gráfico como empleando comandos, será consciente de la importancia de las variables de entorno y será capaz de trabajar con ellas en los principales sistemas operativos y conocerá las principales técnicas de administración de discos, así como las posibilidades de automatización de tareas.

1. ESTRUCTURA DE ALMACENAMIENTO DE LA INFORMACIÓN. SISTEMAS DE ARCHIVOS.

Todas las aplicaciones necesitan almacenar y recuperar información. Por ejemplo, cuando se escribe una carta mediante un procesador de textos hay que almacenar los caracteres escritos para poderlos recuperar cuando se desee.

Mientras una aplicación se está ejecutando, puede almacenar determinada información en el espacio de direcciones, dentro de la memoria RAM, que tiene asignado. Sin embargo, este espacio es demasiado pequeño para algunas aplicaciones, como por ejemplo las bancarias.

Además otro problema que se plantea al mantener la información dentro del espacio de direcciones de un proceso es que cuando el proceso termina la información se pierde.

Por todo lo anterior es necesario usar otro espacio de almacenamiento distinto de la memoria RAM, que nos proporcione mayor capacidad y que sea permanente.

Un sistema de almacenamiento de información debe almacenar gran cantidad de datos, los cuales deben sobrevivir a la terminación del proceso que la usa, y además debe ser accesible a múltiples procesos concurrentes.

El **sistema de archivos** es la parte del sistema operativo encargada de administrar el almacenamiento secundario. Sus funciones más representativas implican la creación, eliminación y modificación de archivos.



Disco Duro



Disquete



Cinta Magnética



Grabadora CD/DVD



Unidad ZIP



Pendrive

Imagen: principales dispositivos de almacenamiento secundario

Para la creación de archivos se requieren dos pasos: en primer lugar hay que encontrar espacio libre en disco y en segundo lugar hay que anotar al nuevo archivo en el directorio.

El sistema de archivos puede abrir archivos para ser utilizados de diferentes formas (escribir, leer, ejecutar) y además se pueden hacer consultas de un archivo con el propósito de modificar el mismo. Lógicamente, el sistema de archivos se encargará también del cierre cuando ya no se trabaje con él.

Un sistema de archivos debe ofrecer otro tipo de ayudas a la gestión como por ejemplo:

Compartición de archivos entre usuarios.

Transferencia de datos entre archivos.

Estructurar los archivos de la forma más adecuada a cada aplicación concreta.

Gestión de los soportes físicos y dispositivos que los manejan.

Seguridad y protección, garantizando la integridad de los datos, los derechos de acceso y condiciones de utilización.

Cada sistema operativo tiene su propio sistema de archivos. A continuación se va a describir brevemente alguno de los principales.

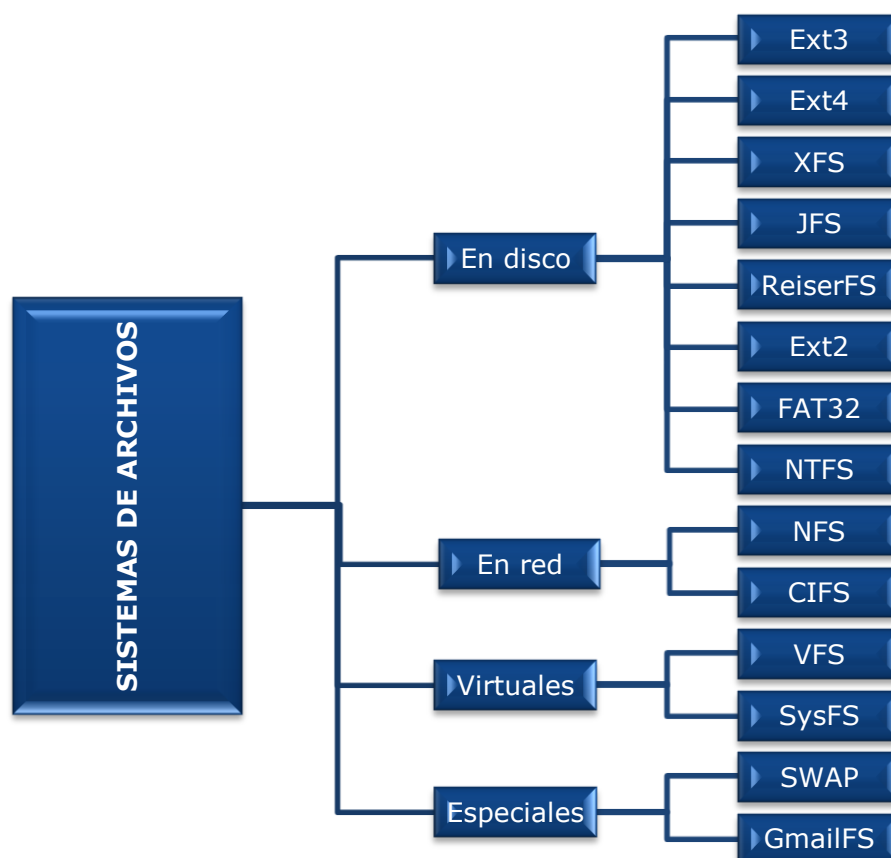


Imagen: principales sistemas de archivos

1.1 Sistemas de archivos Linux

La base es el sistema de archivos extendido (EXT), que utiliza una estructura de metadatos inspirada en UNIX. Este sistema cuenta con diferentes revisiones (ext2, ext3, ext4), que han ido mejorando progresivamente tanto la seguridad como la fiabilidad.

Ext3. Este sistema de archivos se desarrolla tomando como base el popular sistema de ficheros ext2, muy extendido en los sistemas Linux. De hecho, la conversión entre estos 2 sistemas de archivos es sencilla y rápida. La principal mejora que aporta ext3 sobre su predecesor es que incorpora un sistema de journaling (un mecanismo para implementar transacciones). El procedimiento de journaling utilizado en ext3 garantiza la protección tanto de los datos como de los metadatos en caso de caída inesperada del sistema. El principal éxito de este sistema de archivos es la fácil y rápida conversión de ext2 a ext3.

Ext4. Desde la versión 2.6.28 del kernel de Linux se ha añadido esta nueva versión del sistema de archivos que incluye algunas mejoras respecto a la anterior ext3. Ubuntu lo ha añadido como sistema de archivos por defecto a partir de la versión 9.10. Una de las principales mejoras respecto a ext3 es la introducción de los extents, que es un rango de bloques físicos contiguos utilizados para minimizar la fragmentación de los archivos.

1.2 Sistemas de archivos Windows

FAT. Es el más antiguo, y se caracteriza por el uso de una tabla de asignación de archivos (FAT) y clusters. En el sistema de archivos FAT, los clusters son las unidades más pequeñas de almacenamiento de datos; cada uno contiene una cantidad determinada de sectores de disco. FAT se usa para registrar qué clusters se encuentran en uso, cuáles están sin usar, y dónde se ubican los archivos.

Su sucesor es **FAT32**, que se basa en entradas de tabla de asignación de archivos de 32 bits, en lugar de las entradas de 16 bits que usa el sistema de archivos FAT. Como resultado, FAT32 soporta volúmenes mucho mayores

NTFS. El Sistema de Archivos de Nueva Tecnología (NTFS) mejora las características de sus predecesores, tanto a nivel de seguridad como de fiabilidad y escalabilidad.

1.3. Sistemas de archivos en red

NFS (network filesystem): desarrollado inicialmente por Sun Microsystems, suele ser la opción por defecto para sistema de ficheros en red sobre GNU/Linux. El protocolo es independiente de la máquina, del sistema operativo y del protocolo de transporte, ya que implementan CRPC. Es interesante señalar que todas las opciones son síncronas (respecto al trabajo sobre el fichero).

CIFS (common internet filesystem): también conocido como SMB o Samba, la implementación más utilizada es la desarrollada por Microsoft, y es utilizado en sistemas Windows. Permite compartir ficheros e impresoras por la red, y Linux puede implementar tanto la versión de servidor como de cliente. Es decir, permite la convivencia simultánea de sistemas Windows y GNU/Linux en la misma red de área local.

1.4. Sistemas de archivos Macintosh.

Su sistema propio es el sistema de archivos jerárquico (HFS), que ha evolucionado a **HFS Plus** (también conocido como HFS Extended y Mac OS Extended), una versión mejorada que soporta archivos mucho más grandes (ya que emplea bloques direccionables de 32 bits en vez de 16) y que emplea Unicode (en vez de Mac OS Roman) para el nombre de los archivos.



ENLACE DE INTERÉS

En el siguiente enlace se puede ampliar información sobre los sistemas de archivos Macintosh:

<https://es.wikipedia.org/wiki/HFS%2B>

2. ORGANIZACIÓN DE ARCHIVOS

Un **fichero** o **archivo** es un conjunto de información relacionada de forma lógica a la que se da un nombre (nombre del archivo) para identificarla de forma unívoca.

Un aspecto importante a tener en cuenta es que en los ordenadores la información se representa mediante elementos y circuitos electrónicos de dos estados, On, Off. Es por eso que trabajan con **números binarios**, ya que sus circuitos internos pueden representar e interpretar esta dualidad.

2.1. Sistema binario

El **Sistema Binario** es un sistema de numeración de base 2, ya que lo componen dos elementos, $\{0,1\}$. Es un sistema posicional (el valor de un dígito está determinado por la posición que ocupa).

Ejemplo: 1000b, que equivale al número 8 en decimal.

2.1.1. Conversión del sistema binario al decimal.

Siendo:

- $b \rightarrow$ base del sistema de numeración
- $N \rightarrow$ un número de i dígitos.
- $X_i \rightarrow$ coeficientes

Obtenemos el valor decimal mediante el siguiente polinomio:

$$N \rightarrow X_{i-1} * b_{i-1} + X_{i-2} * b_{i-2} + \dots + X_1 * b_1 + X_0 * b_0$$

Ejemplo: Pasar el número 101b, a decimal

$$N=101_2 \rightarrow 1 * 2^2 + 0 * 2^1 + 1 * 2^0 \rightarrow 4 + 0 + 1 \rightarrow 5_{10}$$

Otro método que puede resultar más intuitivo sería utilizar una tabla para la conversión. En dicha tabla:

1. Se representan las potencias de 2 comenzando por la parte derecha. En la primera fila se escriben en modo de potencia, y en la segunda con sus equivalentes valores decimales.
2. Se sitúa el número binario de derecha a izquierda en la fila 3.

3. Se selecciona en la segunda fila los números que se corresponden en la tercera fila con un 1 y se suman.

Ejemplo. Dado el siguiente número binario: 110100b, pasarlo a decimal

...	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
...	128	64	32	16	8	4	2	1
...	0	0	1	1	0	1	0	0
32+16+4= 52. Sumamos donde hay 1								

Imagen: conversión de binario a decimal mediante tabla

Resultado: 110100_b → 52 en decimal

2.1.2. Conversión del sistema decimal a binario

Se divide el número decimal entre la base, 2, y se va obteniendo el resto hasta que el cociente es 0.

El resultado es el conjunto de restos en orden inverso.

Ejemplo: Transformar el número 25 de decimal a binario

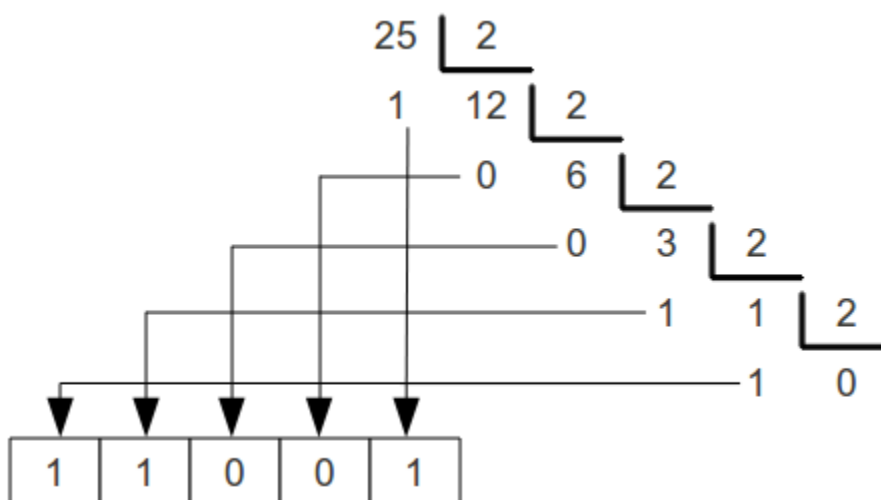


Imagen: conversión de decimal a binario mediante divisiones sucesivas

Resultado: $25_{10} \rightarrow 11001_2$

2.2. Unidades de almacenamiento de datos

El uso del sistema binario condiciona que la mínima unidad de información con la que podemos trabajar en un ordenador sea el **BIT** (BInary digiT). Un bit es una unidad que nos permite almacenar un único dígito binario, un 0 o un 1.

Por agrupación de bits se van formando las unidades superiores, comenzando por el **byte**, que es un conjunto de 8 bits.

Ejemplo: 10010011b.

Se utilizan **prefijos** para establecer unidades múltiplos de los bytes, debido al excesivo tamaño de algunos archivos o la cantidad de éstos, de la manera siguiente:



Imagen: jerarquía de datos

2.3 Jerarquía de datos

En general, los datos están jerarquizados según un modelo general de bits (dígitos binarios, 0 ó 1) que forman bytes o caracteres, que a su vez forman campos, que completan la jerarquía formando registros.

Un archivo es por lo tanto una agrupación de registros relacionados entre sí, cuyo significado lo proporciona el diseñador. La longitud de los registros puede ser fija o variable. La estructura lógica de un fichero puede ser:

- Una secuencia de bits (sin estructura); ésta permite la máxima flexibilidad al usuario.
- Una secuencia de registros de longitud fija; se utilizaba en sistemas operativos primitivos como CP/M. Hoy día no se utiliza.
- Un árbol de registros de longitud variable, donde cada registro tiene una clave.

Un archivo tendrá una estructura general de acuerdo con su tipo. Por ejemplo, un archivo de texto es una secuencia de caracteres organizados en líneas.



ENLACE DE INTERÉS

En la siguiente web puede ver una infografía muy completa sobre la historia de los dispositivos de almacenamiento y sus capacidades:

<https://www.revistacloudcomputing.com/2013/08/infografia-de-la-evolucion-del-almacenamiento-informatico/>

2.4. Características de los archivos

Los sistemas de archivos usan los ficheros como **abstracción** para no hacer al usuario preocuparse de cómo y dónde se almacena físicamente la información en disco.

La característica más importante es la abstracción, la forma de nombrar los objetos que se manejan. Cuando un proceso crea un archivo le da un nombre y cuando acaba otros procesos pueden hacer uso de él mediante ese nombre.

Las **reglas de nombrado** de ficheros varían de unos sistemas operativos a otros. Unos usan un número de caracteres determinados, otros omiten en los nombres varios caracteres, algunos distinguen mayúsculas y minúsculas, otros no, etc...

Algunos sistemas operativos dividen el nombre del fichero en **nombre.extensión**. La razón de esta división es sobre todo identificar el tipo de archivo en cuestión, y poder relacionarlo con facilidad con el/los programas que lo manejan. Así por ejemplo en los sistemas operativos Windows, algunas de las extensiones más conocidas están asociadas a programas muy extendidos, mientras que otras son un estándar como por ejemplo:

Tipos de archivo	Extensión asociada
Archivos de texto	Txt
Archivos de texto enriquecido	Rtf
Archivos ejecutables (aplicaciones)	Exe
Archivos gráficos	Jpg Gif Bmp
Documentos de Word	Doc, Docx
Documentos de Excel	Xls, Xlsx
Bases de Datos de Access	Mdb
Documentos de Adobe Acrobat	Pdf
Archivos comprimidos	Zip, Rar, Ace



ENLACE DE INTERÉS

En el siguiente enlace se muestra una lista más completa de extensiones de archivos y los programas que trabajan con ellas

http://es.wikipedia.org/wiki/Extensi%C3%B3n_de_archivo

Éstas son sólo algunas de las extensiones más comúnmente extendidas. Algunos programas manejan archivos en un formato específico y por tanto estos archivos solamente pueden ser abiertos por dichos programas, siendo necesaria la instalación de los mismos para poder tratarlos. También puede darse el caso de que programas diferentes utilicen la misma extensión, siendo en este caso necesario cargar el programa y desde este abrir el

archivo, ya que el sistema operativo solamente puede asociar la extensión de un archivo con un programa en concreto.

Por ejemplo, un archivo gráfico se puede abrir con el programa 'Visor de imágenes de Windows' o bien con un programa de retoque fotográfico, como 'Paint'. Asimismo, con el programa 'Paint', no podemos abrir archivos de texto, ya que éstos están compuestos de una secuencia de caracteres de texto, no comprensibles por dicho programa y no de una secuencia de bytes que codifican un archivo gráfico.

Por otra parte, muchos usuarios de Windows creen que el hacer doble clic sobre un archivo se carga el archivo en la aplicación correspondiente, sin saber que dicho archivo puede ser abierto con otros programas. Para ello basta con pulsar dicho archivo con el botón derecho del ratón y elegir la opción "Abrir con".

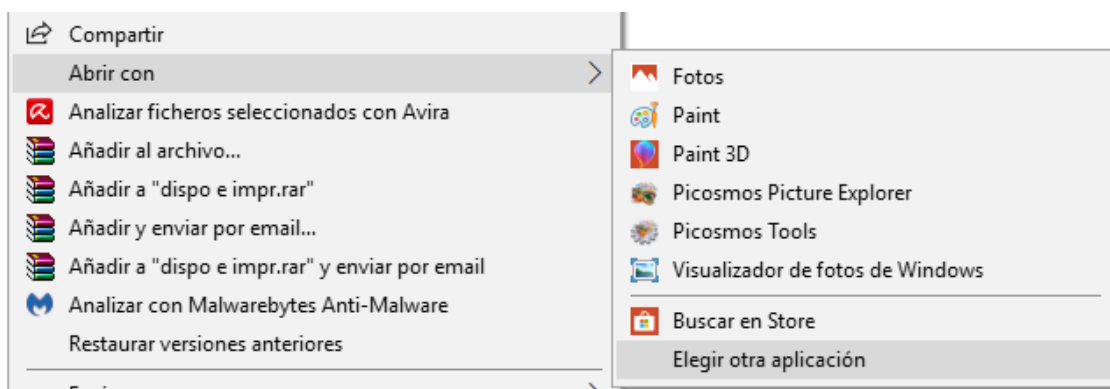


Imagen: opción Abrir con del menú contextual en Windows 10

En el panel que se abre a continuación podemos elegir el programa deseado entre los que se nos ofrecen o buscar el programa deseado entre los que haya instalados en el equipo. Además, podemos marcar una casilla para que en adelante los archivos de este mismo tipo se abran siempre con el programa elegido.

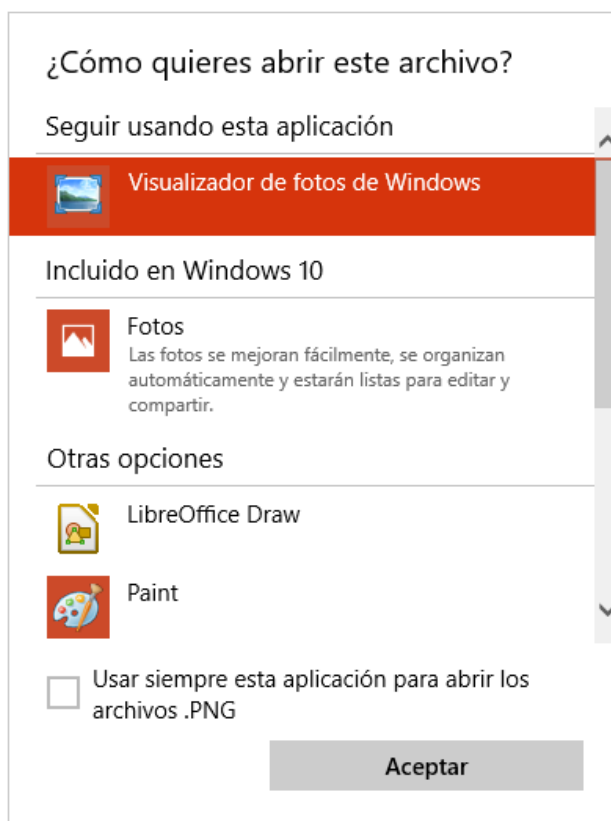


Imagen: selección del programa para abrir un archivo

2.5. Tipos de archivos

Se diferencian básicamente tres tipos de archivos, los regulares, los de dispositivos y los directorios.

- Los **ficheros regulares** almacenan información del usuario, en binario o código ASCII
- Los **ficheros de dispositivos** (de bloques o de caracteres) hacen que se pueda tratar de una forma uniforme a los dispositivos mediante las operaciones definidas sobre los ficheros.
- Los ficheros **directorio** almacenan información sobre la estructura del propio sistema de ficheros, guardando información sobre otros ficheros o directorios. Gracias a los directorios podemos tener estructuras jerárquicas en los sistemas de ficheros.

3. FICHEROS Y DIRECTORIOS

A continuación vamos a revisar las principales operaciones que se pueden llevar a cabo tanto con ficheros como con directorios.

3.1 Operaciones con ficheros

Un sistema de ficheros suele ofrecer las siguientes operaciones sobre archivos:

Create.

- El archivo se crea sin datos. El propósito de la llamada es anunciar que va a haber un archivo y establecer algunos de los atributos.

Delete.

- Cuando el archivo ya no se necesita, es preciso eliminarlo para desocupar el espacio en disco. Siempre hay una llamada al sistema para este fin.

Open.

- Antes de usar un archivo, un proceso debe abrirlo. El propósito de la llamada Open es permitir al sistema que obtenga los atributos y la lista de direcciones de disco. Es decir, el sistema se prepara para hacer uso del fichero.

Close.

- Una vez concluidos todos los accesos, los atributos y las direcciones de disco ya no son necesarios, por lo que se debe cerrar el archivo para liberar el espacio correspondiente en las tablas internas. Muchos sistemas fomentan esto limitando a los procesos a un número máximo de archivos abiertos.

Read.

- Se leen datos del archivo. Por lo regular, los bytes leídos provienen de la posición actual dentro del fichero abierto. Es decir, si en una operación anterior se había leído hasta un determinado byte, en la siguiente lectura se comienza a leer desde la posición donde anteriormente nos habíamos quedado. El invocador debe especificar cuántos datos se necesitan y también debe suministrar un buffer para colocarlos.

Write.

- Se escriben datos en el archivo, también por lo general, en la posición actual como ocurría con la lectura. Si dicha posición está al final del archivo, el tamaño del archivo aumenta en tantos bytes como contengan los datos escritos. Si la posición actual no está al final del archivo, se sobrescriben en los datos existentes.

Append.	<ul style="list-style-type: none"> •Esta llamada es una forma restringida de Write que solo puede agregar datos al final del archivo. Los sistemas que ofrecen un juego mínimo de llamadas al sistema generalmente no cuentan con Append.
Seek.	<ul style="list-style-type: none"> •En el caso de archivos de acceso aleatorio, se requiere de un método para especificar de donde extraer los datos. Un enfoque común es tener una llamada al sistema, que ajuste el apuntador a la posición actual haciéndole que apunte a un lugar específico del archivo. Una vez efectuada esta llamada, se pueden leer datos de esa posición o escribirlos en ella.
Get Attributes.	<ul style="list-style-type: none"> •Es frecuente que los procesos necesiten leer los atributos de un archivo para realizar su trabajo. Para eso existe esta llamada que se encarga de leerlos y devolvérselos al proceso que los haya solicitado.
Set Attributes.	<ul style="list-style-type: none"> •Algunos de los archivos pueden ser establecidos por el usuario y modificarse después de que se creó el archivo. Esta llamada al sistema lo hace posible. Por ejemplo se usa para modificar los flags del fichero.
Rename.	<ul style="list-style-type: none"> •Es común que un usuario necesite cambiar el nombre de un archivo existente. Esta llamada permite hacerlo, aunque no siempre es indispensable, ya que el archivo por lo regular puede copiarse en un archivo nuevo con el nombre, eliminando después el viejo.

3.2 Operaciones con directorios

Los directorios tienen un papel importantísimo en los sistemas de ficheros. Un **directorio** contiene varias entradas, una por archivo que hay en su interior. Cada una de esas entradas contiene información relativa al fichero. Por ejemplo el nombre y sus atributos.

Cuando se solicita la apertura de un archivo a un sistema de ficheros, este busca ese archivo en un determinado directorio indicado en la petición hasta encontrar el nombre del archivo y luego extraen los atributos y el contenido del mismo en disco.

El tener una jerarquía permite por ejemplo que varios usuarios puedan tener archivos con el mismo nombre ya que cada uno los tendrá en su directorio correspondiente.

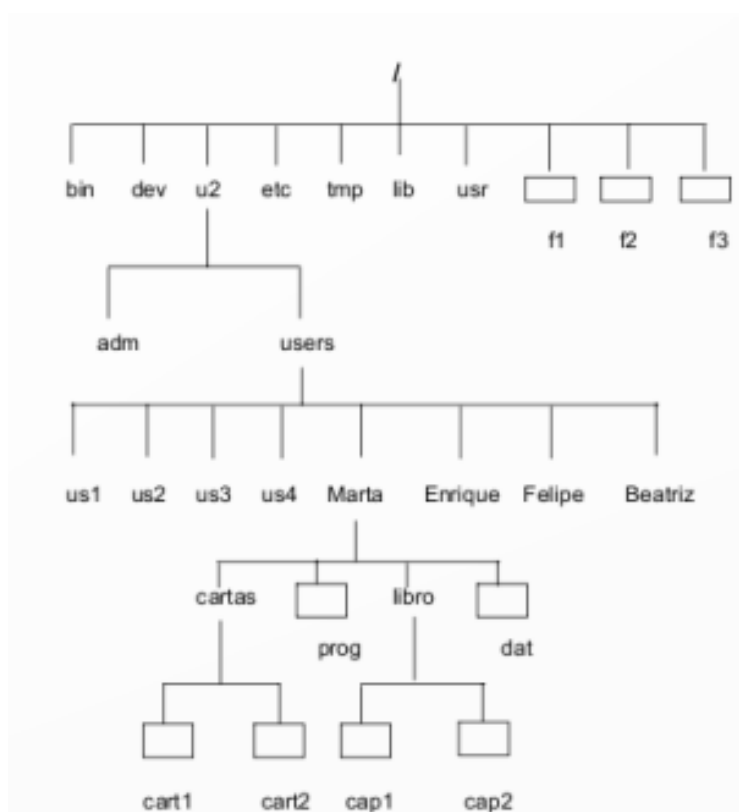
Al hablar de los directorios, aparece un nuevo concepto relativo a los sistemas de ficheros. Se trata del **pathname o ruta de acceso**, que es la forma de nombrar o especificar dónde está el fichero o directorio en la jerarquía de archivos del dispositivo de almacenamiento.

Existen dos métodos para especificar la ruta:

- **Ruta absoluta:** se especifica el camino para llegar al fichero partiendo desde el directorio raíz (el origen de la jerarquía).
- **Ruta relativa:** se especifica el camino partiendo del directorio de trabajo o directorio actual.

EJEMPLO PRÁCTICO

Dada la siguiente estructura de archivos



Queremos acceder al directorio cap2 de la usuaria Marta. ¿De qué formas puede hacerse?

Solución:

- Mediante una ruta absoluta: /u2/users/Marta/libro/cap2
- Mediante rutas relativas, como:
 - Desde el directorio Marta : libro/cap2
 - Desde el directorio users : Marta/libro/cap2

Al igual que con los archivos normales, con los directorios también se pueden realizar distintas operaciones. Las más comunes son:

Create.	<ul style="list-style-type: none"> • Se crea un directorio, que este vacío con la excepción de punto (.) y punto punto (..), que el sistema coloca automáticamente.
Delete.	<ul style="list-style-type: none"> • Se elimina un directorio. Solo puede eliminarse un directorio vacío. Un directorio que solo contiene punto y punto punto se considera vacío, ya que estos normalmente no pueden eliminarse.
Opendir.	<ul style="list-style-type: none"> • Los directorios pueden leerse. Por ejemplo, para listar todos los archivos de un directorio, un programa para emitir listados abre el directorio y lee los nombres de los archivos que contiene. Antes de poder leer un directorio, es preciso abrirlo, de forma análoga a cómo se abren y leen los archivos.
Closedir.	<ul style="list-style-type: none"> • Una vez que se ha leído un directorio, debe cerrarse para liberar espacio de tablas internas.
Readdir.	<ul style="list-style-type: none"> • Esta llamada devuelve la siguiente entrada de un directorio abierto. Antes, era posible leer directorios empleando la llamada al sistema Read normal, pero ese enfoque tiene la desventaja de obligar al programador a conocer y manejar la estructura interna de los directorios. En contraste, Readdir siempre devuelve una entrada en un formato estándar, sin importar cual de las posibles estructuras de directorios se esté usando.
Rename.	<ul style="list-style-type: none"> • En muchos sentidos, los directorios son iguales que los archivos y podemos cambiar su nombre tal como hacemos con los archivos.

3.3 Estructura de directorios en Linux

Ya se ha visto que la información en los sistemas de almacenamiento se guarda en archivos y éstos a su vez, se pueden agrupar en directorios y subdirectorios formando una estructura en árbol. En los sistemas Windows, cada unidad de almacenamiento se nombra con una letra (C:, D:, E:, F:...) y constituye su propia estructura de directorios que parte de un directorio conocido como directorio raíz. Las unidades de almacenamiento pueden ser los discos duros o particiones, unidades ópticas, discos externos USB o en general cualquier dispositivo que pueda almacenar información. Sin

embargo, en Linux existe **un solo árbol de directorios** y cualquier dispositivo de almacenamiento debe formar parte de él. Así mismo, existe un solo directorio raíz que estará asociado a un solo dispositivo de almacenamiento, normalmente una partición del disco duro.

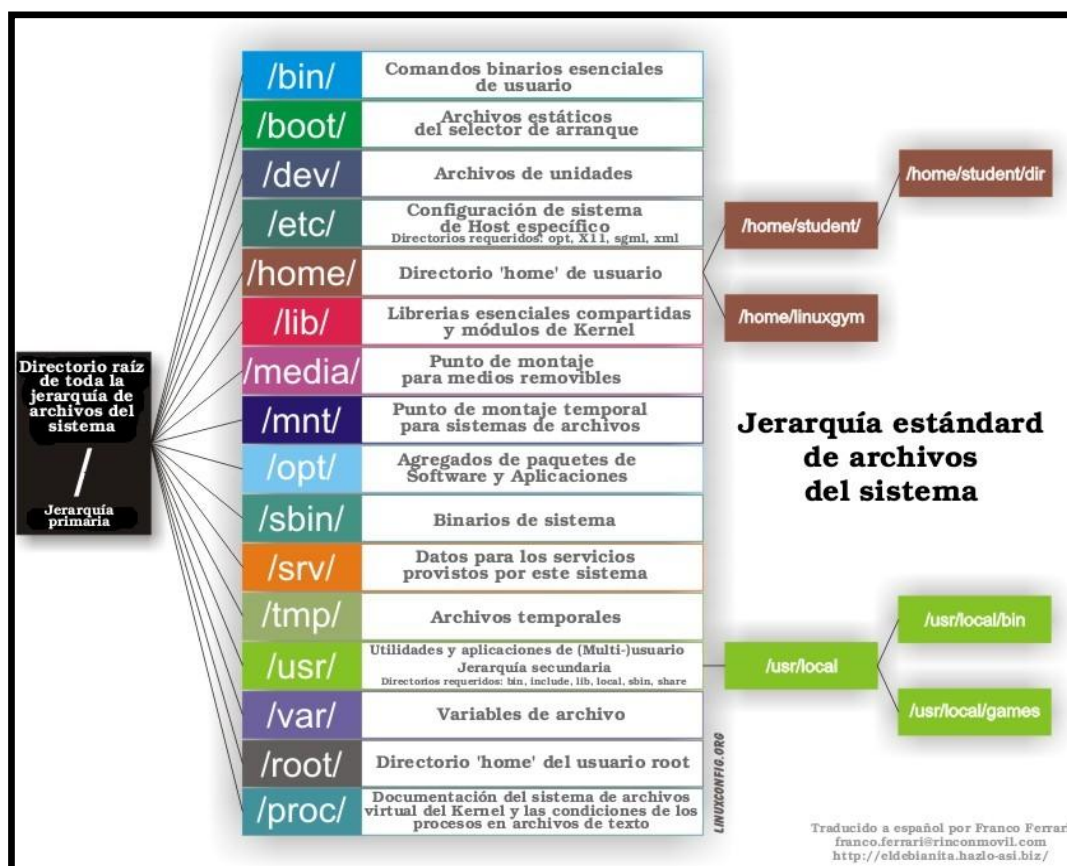


Imagen: Estructura de directorios Linux

A la operación de asociar un dispositivo de almacenamiento a un directorio del sistema de archivos se le conoce como **montaje de dispositivos**.

Hay que indicar que en Linux el directorio raíz se representa por el carácter / que también se utiliza como separador de directorios. En Windows se utiliza la barra invertida \.

Por tanto, toda ruta que comience por el carácter / significa que es una ruta absoluta, es decir que parte del directorio raíz. Como se ha visto, en los sistemas Linux las ubicaciones de los archivos en el sistema siguen unas normas determinadas con el objeto de aumentar el nivel de organización.

La mayoría de las distribuciones de Linux siguen el llamado estándar de jerarquía del sistema de archivos, **FHS** (Filesystem Hierarchy Standard).

Algunos de los directorios más importantes son los siguientes:

/bin y /usr/bin. Estos directorios contienen la mayoría de los archivos ejecutables y comandos más comunes del sistema Linux.

/sbin y /usr/sbin. Estos directorios también contienen comandos y archivos ejecutables normalmente usados en tareas de administración, con lo cual muchos de ellos sólo son ejecutables para el usuario root o administrador del sistema.

/etc. Este directorio contienen los archivos de configuración de todo el sistema, por lo que es de vital importancia. Normalmente contiene archivos de configuración de tipo texto sin formato. Además suele tener permisos de lectura de modo que sólo el usuario root (administrador) los puede modificar.

/root. Directorio home del usuario root.

/usr. Directorio destinado a almacenar las aplicaciones, con lo cual su tamaño puede ser elevado si existen muchos paquetes de software instalados.

/home. Directorio donde se almacenan todos los directorios homes de los usuarios del sistema, por tanto, en función del número de usuario y de uso que hagan del sistema, este directorio puede llegar a necesitar mucho espacio también. En sistemas servidores es habitual y recomendable utilizar una partición separada para este directorio.

/lib y /usr/lib. Directorios que contienen librerías compartidas del sistema.

/tmp. Directorio para almacenar ficheros temporales.

/boot. Directorio que contiene los archivos necesarios para el arranque del sistema. Por ejemplo, aquí se almacenan los archivos del gestor de arranque si hubiera alguno instalado. También se suele almacenar las imágenes del kernel. Algunas distribuciones aconsejan utilizar una pequeña partición separada para este directorio.

/dev. Directorio que almacena ficheros de dispositivos. Estos ficheros no son realmente tales, sino que es la forma en la que los sistemas Linux implementan los controladores de dispositivos.

/var. Directorio que contiene información variable en general como colas de impresión, colas de envío y recepción de correos, archivos de registro y de eventos del sistema... En sistemas Linux utilizados como servidores, este directorio puede necesitar mucho espacio, por lo cual es recomendable utilizar una partición propia.

/opt. Directorio opcional donde se pueden instalar aplicaciones, a parte de como hemos visto en /usr. En algunos sistemas Linux no existe.

/mnt y /media. Directorios usados por el sistema para realizar el montaje de otros dispositivos de almacenamiento como disquetes, CD-ROM, unidades de almacenamiento USB, etc...

4. GESTIÓN DE ARCHIVOS MEDIANTE COMANDOS Y ENTORNOS GRÁFICOS

La operativa habitual de los archivos y directorios puede llevarse a cabo tanto mediante la utilización de comandos como a través de entornos gráficos que, aunque menos veloces, hacen más intuitivo el trabajo, sobre todo para usuarios no profesionales. A continuación se revisan ambas técnicas tanto para sistemas operativos libres como propietarios.

4.1 Sistemas Windows

Tenemos dos alternativas para realizar la gestión de archivos en sistemas Windows: el uso de comandos o de un entorno gráfico. Vamos a describir ambas a continuación.

4.1.1 Utilización de comandos. *Símbolo del sistema*

Para gestionar los archivos de un equipo mediante comandos en un sistema Windows, lo primero es acceder al **Símbolo del Sistema**.

En Windows 10 podemos hacerlo escribiendo "cmd" en el campo de búsqueda junto al botón de Inicio.

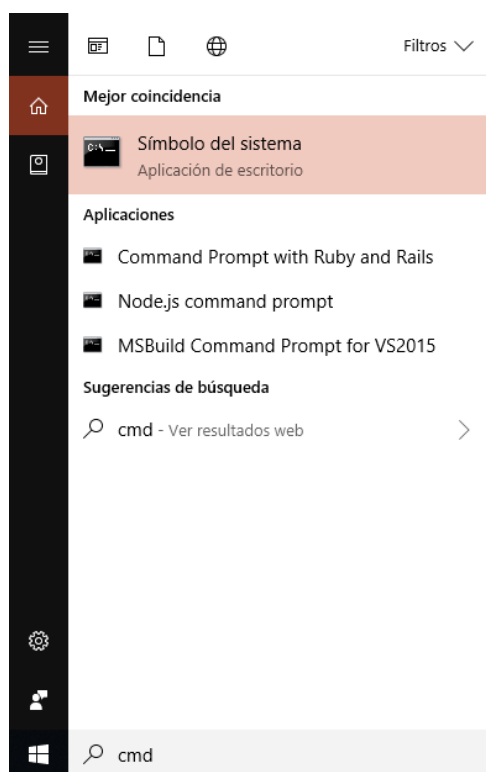


Imagen: acceso al Símbolo del sistema

Al entrar en la aplicación Símbolo de sistema, se encuentra la típica consola de comandos:

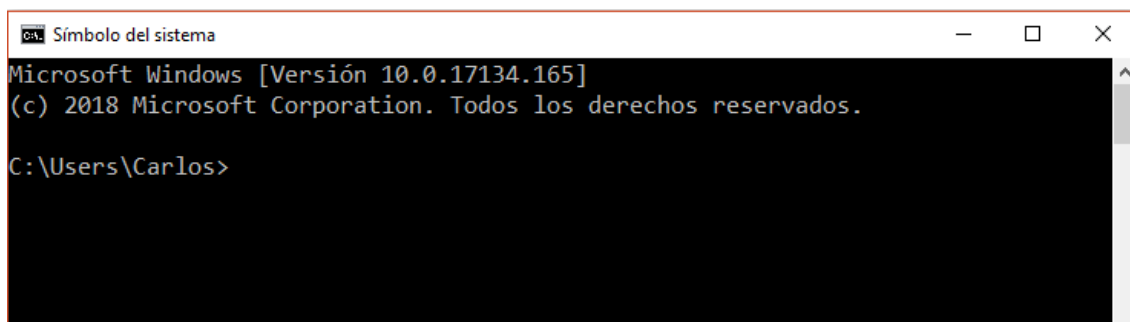


Imagen: herramienta Símbolo del sistema

Como vemos, al entrar a un directorio en Windows mediante comandos, observamos dos directorios creados por defecto:

- 1) '.' → Directorio activo
- 2) '..' → Directorio de nivel superior o padre.

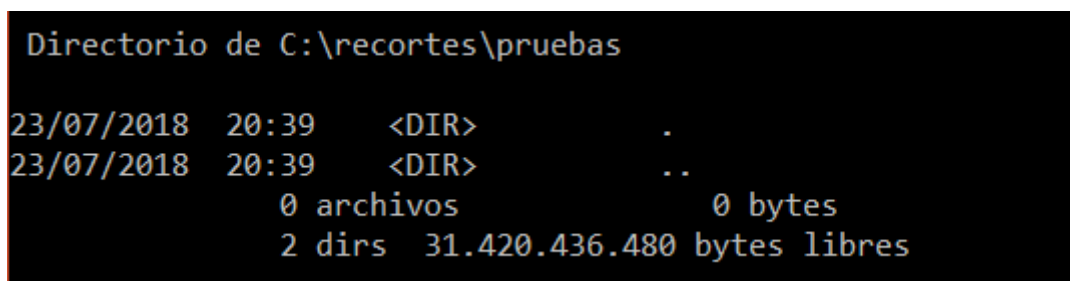
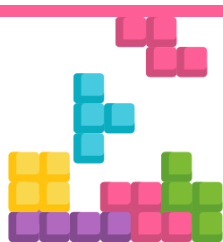


Imagen: directorios por defecto

Una vez dentro de la consola, algunas de las operaciones que se pueden realizar sobre archivos y directorios en Windows son las siguientes:

Comando	Funcionalidad	Ejemplo de utilización
Find	Encontrar una cadena de caracteres en un fichero	Find "José Ramón" agenda.txt
Mkdir (MD)	Crea un directorio en la raíz	MD Clientes
Rename (REN)	Cambia el nombre de fichero	REN agenda.txt agenda.scr
Rd	Borra un directorio	RD cesur
Type	Muestra el contenido de un fichero	Type nombreFichero
Move	Mueve uno o varios archivos a un directorio	Move *.txt c:\temp
Attrib	Permite ver y cambiar los atributos de un archivo	Attrib +r -h fichero.txt Pone al archivo de sólo lectura y le quita el modo oculto
CD	Permite acceder a un directorio	CD Windows
Chkdsk	Analiza el estado de una unidad o de un fichero	Chkdsk C: /f Analiza la unidad C:, detectando y corrigiendo errores
Cls	Limpia la pantalla	No tiene parámetros
Copy	Copia un fichero de un lugar a otro.	Copy actividad.doc C:\temp\
Delete (DEL)	Borra un fichero	Del Sergio.txt Del *.jpg Del temp*.*
Dir	Muestra el contenido de la ruta del directorio especificada	Dir C:\ Muestra el contenido del directorio raíz
Tree	Muestra estructura directorios forma árbol	No tiene parámetros



EJEMPLO PRÁCTICO

Se nos pide crear una carpeta llamada alumnos, y dentro de ella otra llamada primero, que a su vez contenga otras dos carpetas llamadas asignaturas y calificaciones. Una vez creadas todas las carpetas, se solicita mostrar la jerarquía obtenida en forma de árbol.

Solución:

```
Símbolo del sistema

C:\recortes\pruebas>md alumno

C:\recortes\pruebas>cd alumno

C:\recortes\pruebas\alumno>md primero

C:\recortes\pruebas\alumno>cd primero

C:\recortes\pruebas\alumno\primero>md asignaturas

C:\recortes\pruebas\alumno\primero>md calificaciones

C:\recortes\pruebas\alumno\primero>cd..

C:\recortes\pruebas\alumno>tree
Listado de rutas de carpetas para el volumen Windows
El número de serie del volumen es 2C88-B8EF
C:.
├── primero
│   ├── asignaturas
│   └── calificaciones
```

4.1.2 Entorno gráfico. Explorador de Windows

La herramienta gráfica principal para la gestión de archivos y directorios en Windows es el **Explorador de Windows**.

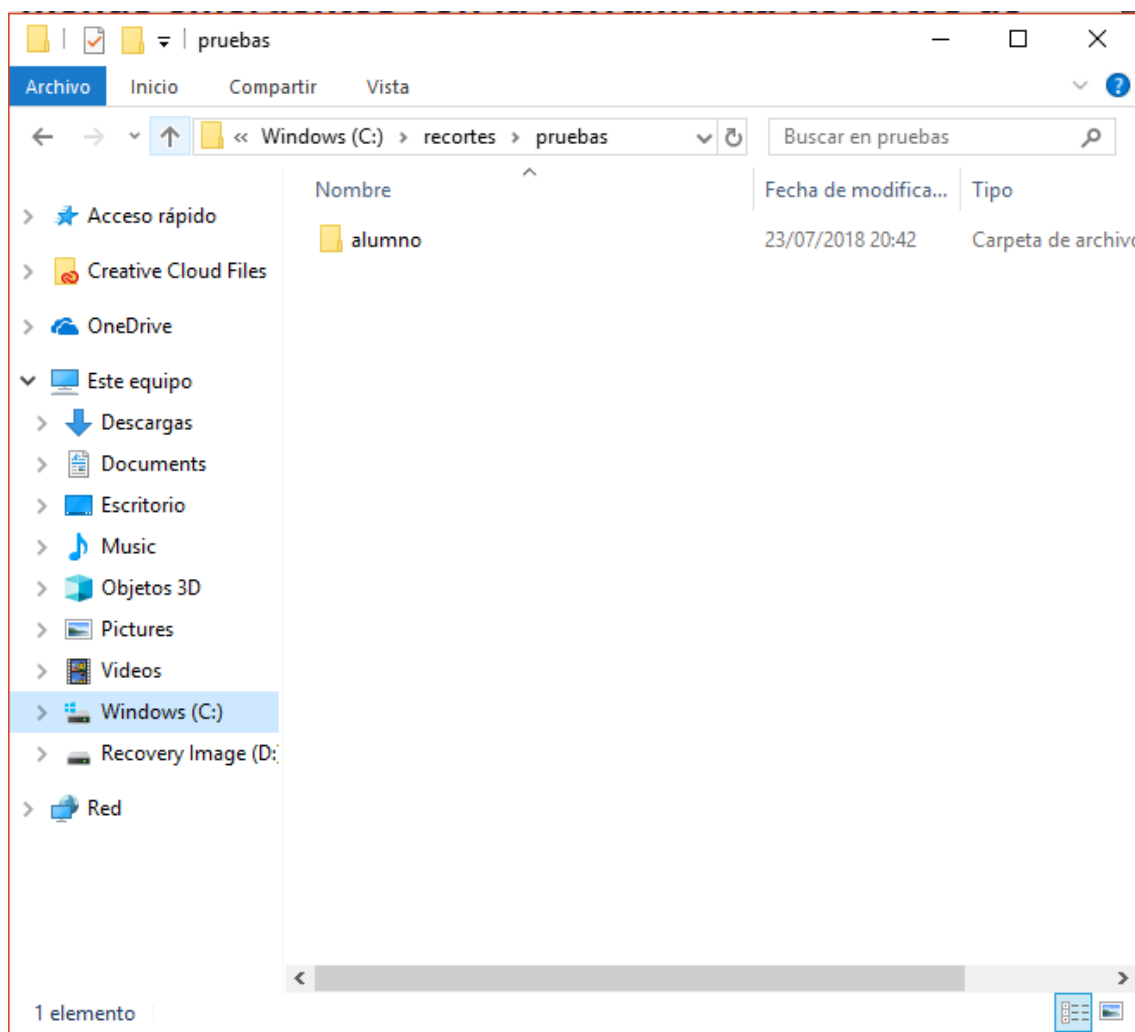


Imagen: explorador de Windows

A continuación se describirá el modo de realizar las principales operaciones con archivos y directorios con dicho explorador.

Para **crear una carpeta** se hace clic con el botón derecho del ratón sobre un espacio libre → aparece un menú → pulsamos Nuevo → Carpeta.

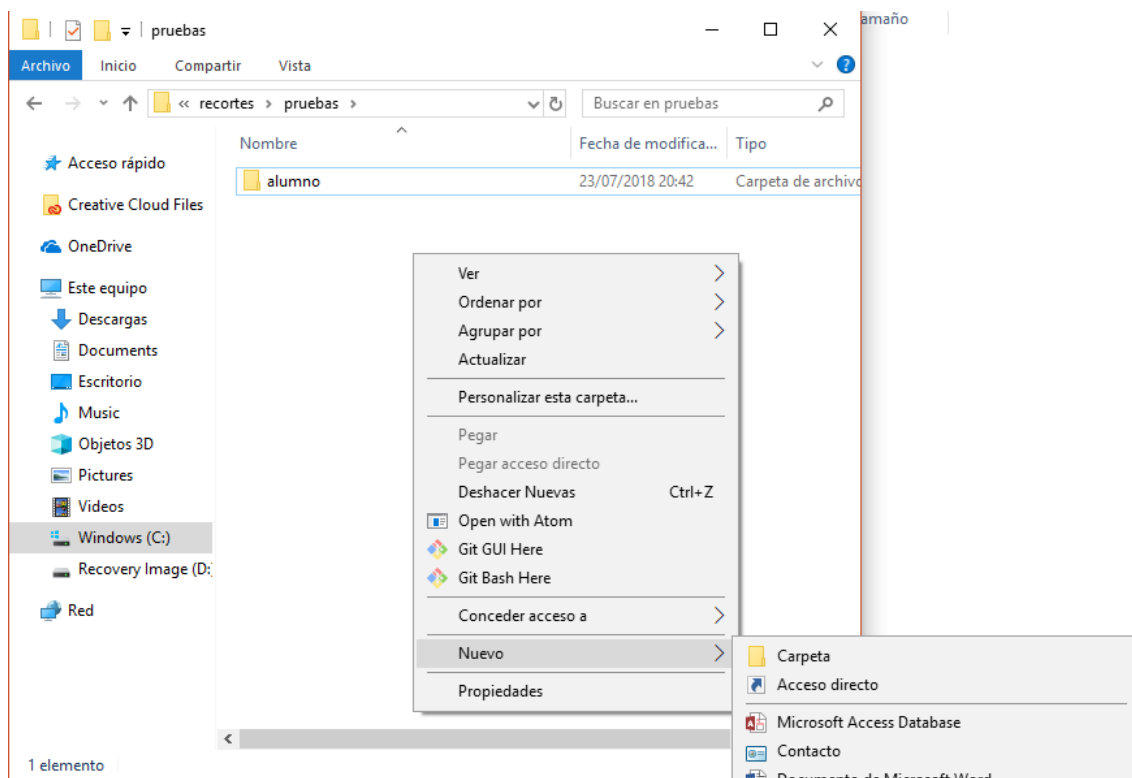


Imagen: crear un directorio/carpeta mediante el explorador de Windows

Para **cambiar el nombre** de la carpeta creada → botón derecho sobre dicha carpeta → Cambiar nombre y escribimos el que queramos

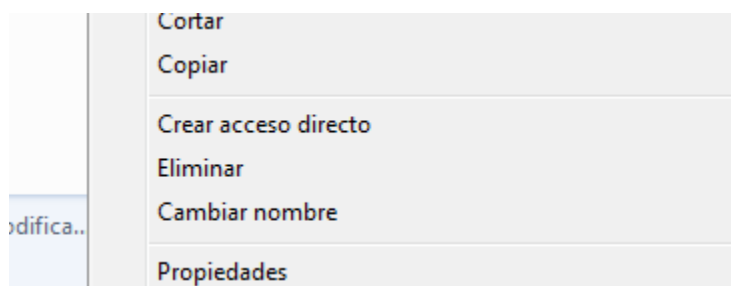


Imagen: cambiar nombre a una carpeta

Una vez cambiado el nombre, para **abrir**la, hacemos doble clic sobre ella o seleccionamos la carpeta → botón derecho → Abrir

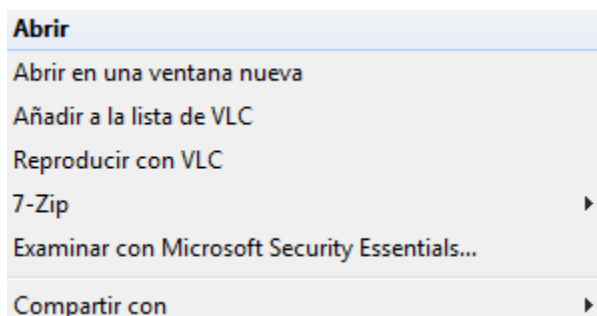


Imagen: abrir una carpeta

Por último, para **eliminar** una carpeta o archivo, hay que seleccionarlo y pulsar la tecla Supr o abrir el menú contextual y seleccionar Eliminar

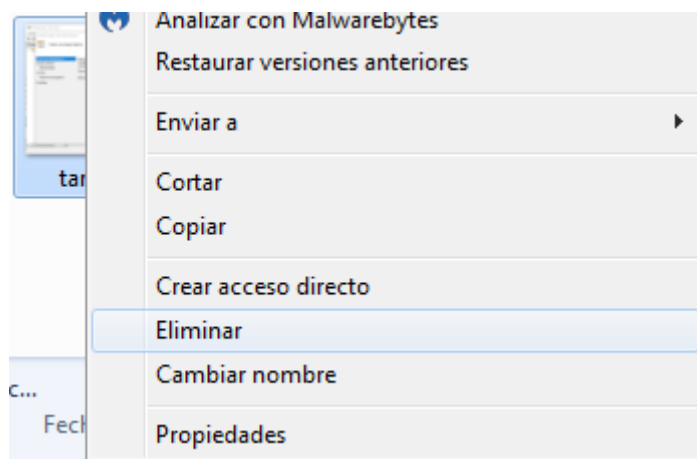


Imagen: eliminar una carpeta

4.2 Sistemas Linux

En sistemas Linux se puede llevar a cabo la gestión de archivos de forma muy similar a la descrita anteriormente para Windows. Se van a realizar los ejemplos sobre la distribución Ubuntu instalada en la unidad anterior, aunque no habría demasiadas diferencias en cualquier otra de las distribuciones principales.

4.2.1 Utilización de comandos. Terminal

Igual que en Windows, lo primero es acceder a la consola de comandos, denominada en este caso Terminal. Se hace mediante la opción de buscar de Unity (el primer icono).

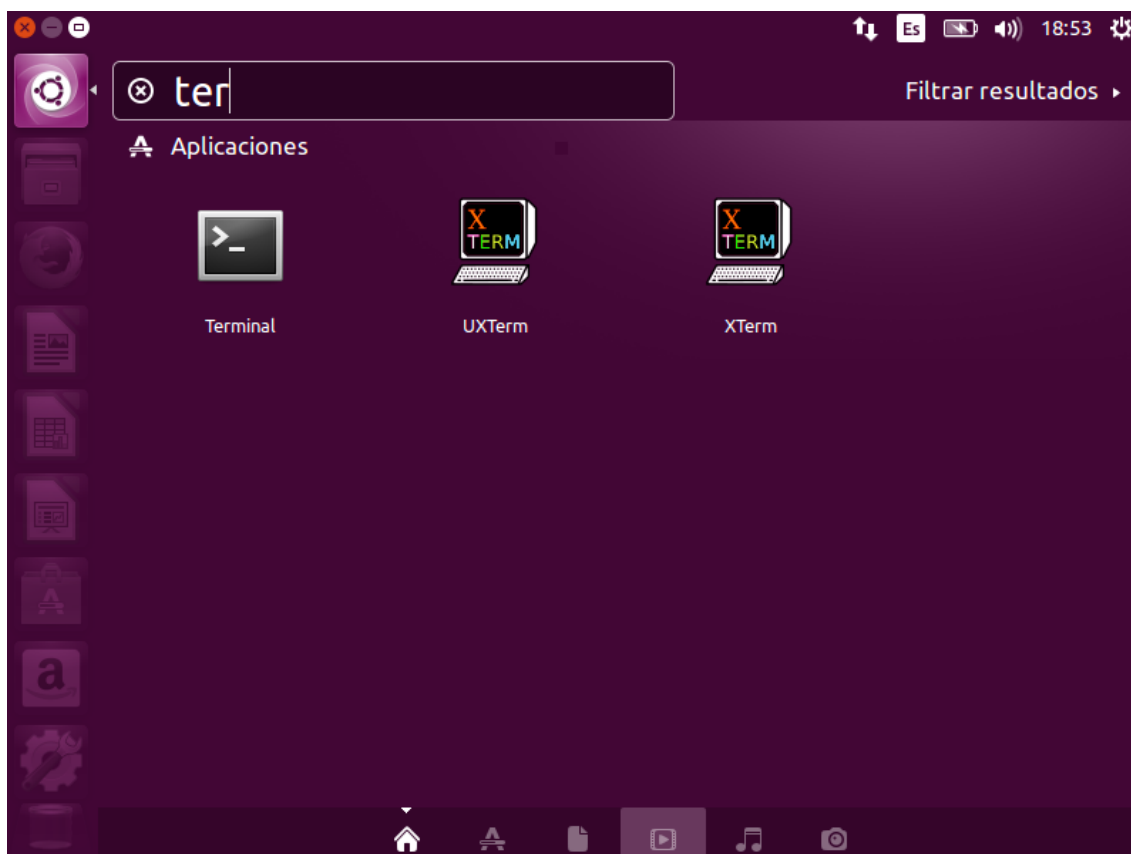


Imagen: acceso al Terminal en Ubuntu

Esta es la herramienta Terminal, lista para recibir órdenes:

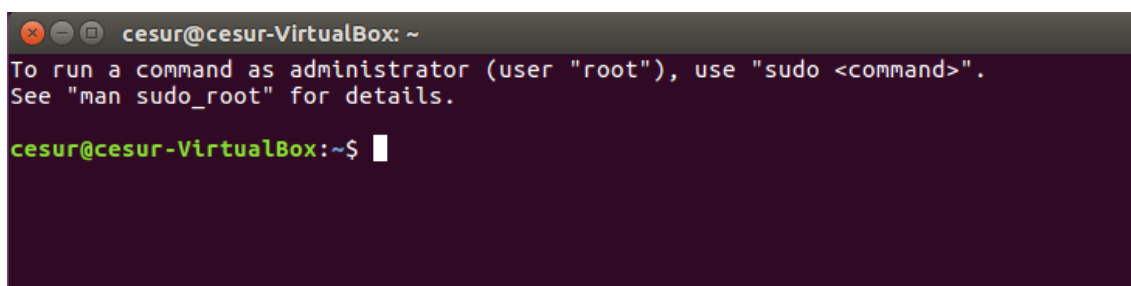
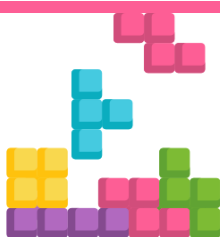


Imagen: Terminal de Ubuntu

Como puede verse, el Terminal nos da acceso al sistema mediante el usuario con el que nos hemos identificado, por lo que se dispone únicamente de los permisos concedidos para este usuario.

Una vez dentro del Terminal, algunas de las operaciones que se pueden realizar sobre archivos y directorios en Linux mediante comandos son las siguientes:

Comando	Funcionalidad	Ejemplo de utilización
Ls	Listar el contenido de un directorio	ls -l
Mkdir	Crear directorio	mkdir trabajos
Cd	Cambiar de directorio	cd trabajos
pwd	Recordar cuál es el directorio actual	pwd
Chmod	Cambiar los permisos de un directorio o archivo	chmod 777 trabajos
Touch	Crear un archivo nuevo en el directorio actual	touch carta
Rmdir	Borrar un directorio vacío	rmdir trabajos
Rm	Borrar un fichero o directorio	rm informe
Mv	Mover o renombrar un archivo	mv carta informe
Cp	Copiar un fichero	cp carta informe
Cat	Mostrar el contenido de un fichero	cat informe



EJEMPLO PRÁCTICO

Se nos pide crear una carpeta llamada documentos, dentro de ella otra llamada trabajo que a su vez contenga otras dos carpetas llamadas cartas e informes. También hay que crear un archivo llamado personal dentro de la carpeta cartas y hacerle una copia, dándole el nombre de privado.

Solución:

```
cesur@cesur-VirtualBox: ~/documentos/trabajo/cartas
cesur@cesur-VirtualBox:~$ mkdir documentos
cesur@cesur-VirtualBox:~$ cd documentos
cesur@cesur-VirtualBox:~/documentos$ mkdir trabajo
cesur@cesur-VirtualBox:~/documentos$ cd trabajo
cesur@cesur-VirtualBox:~/documentos/trabajo$ mkdir cartas
cesur@cesur-VirtualBox:~/documentos/trabajo$ mkdir informes
cesur@cesur-VirtualBox:~/documentos/trabajo$ cd cartas
cesur@cesur-VirtualBox:~/documentos/trabajo/cartas$ touch personal
cesur@cesur-VirtualBox:~/documentos/trabajo/cartas$ cp personal privado
cesur@cesur-VirtualBox:~/documentos/trabajo/cartas$ ls -l
total 0
-rw-rw-r-- 1 cesur cesur 0 ago 30 12:55 personal
-rw-rw-r-- 1 cesur cesur 0 ago 30 12:59 privado
cesur@cesur-VirtualBox:~/documentos/trabajo/cartas$
```

4.2.2 Entorno gráfico. Archivos

La herramienta gráfica principal para la gestión de archivos y directorios en Ubuntu se denomina **Archivos**, y es similar al Explorador de Windows, tanto en aspecto como en funcionalidad. Se accede a ella mediante el segundo icono de la barra de herramientas principal.

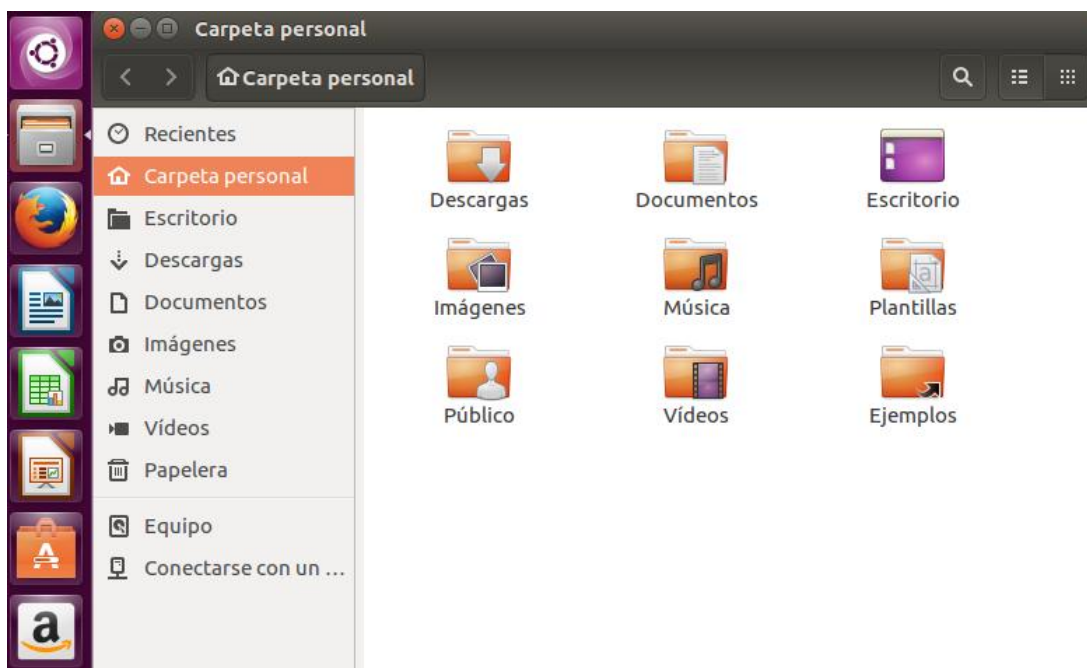


Imagen: herramienta Archivos en Ubuntu

Como se ve en la imagen, se puede navegar por los diferentes directorios del sistema (comenzamos siempre en la **Carpeta personal**), y realizar las principales operaciones usando el menú contextual asociado al botón derecho del ratón.

Por ejemplo, **para crear una carpeta** nueva se hace clic con el botón derecho del ratón sobre un espacio libre → aparece un menú → se pulsa Carpeta nueva.



Imagen: creación de carpeta en Ubuntu

Se asigna un nombre y ya está creada la carpeta

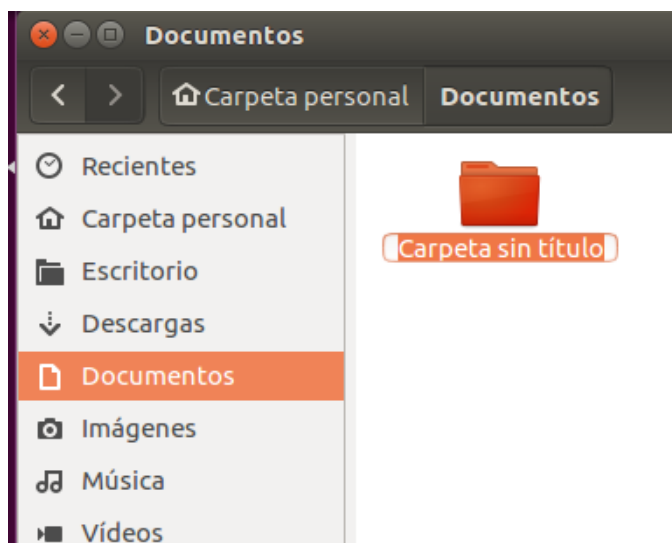


Imagen: asignación de nombre a la carpeta nueva

Para **cambiar el nombre** de una carpeta → botón derecho sobre dicha carpeta → Renombrar, y se escribe el deseado

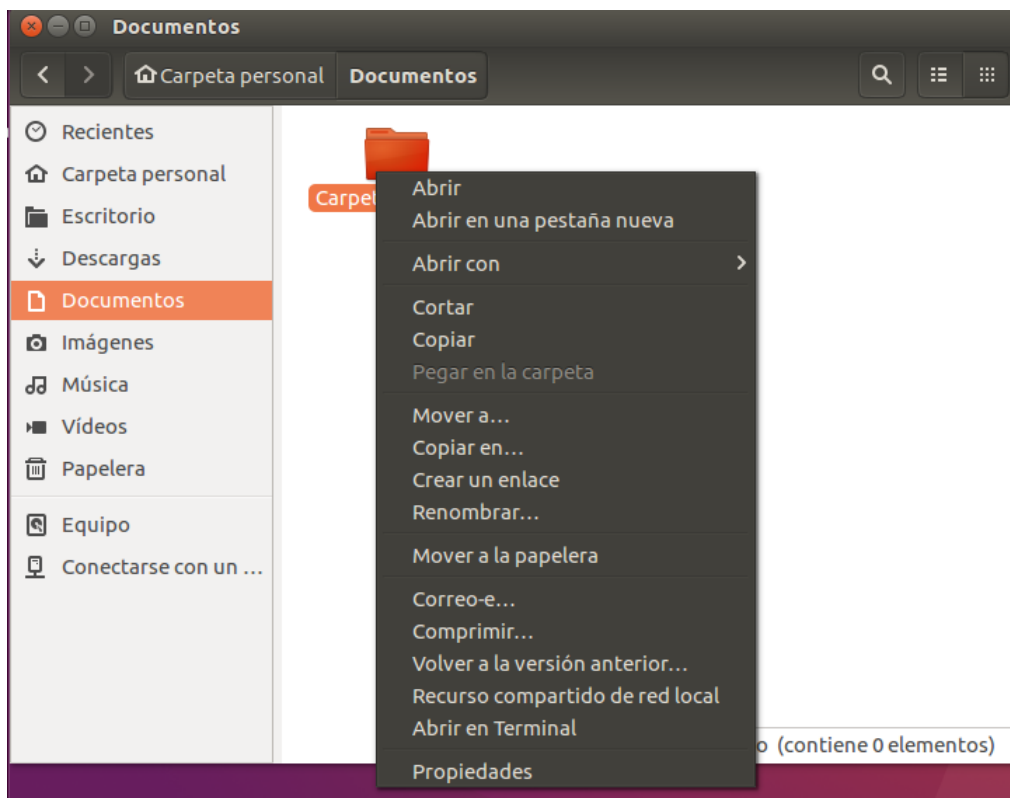


Imagen: renombrar carpeta

El resto de operaciones del menú contextual (Cortar, Copiar, Mover a la papelera para borrar...) nos permiten realizar las operaciones ya comentadas mediante comando, y son bastante intuitivas de llevar a cabo.

5. VARIABLES DE ENTORNO

En un sistema operativo se llama **entorno** o *environment* a un área particular de memoria que almacena variables con información acerca de dicho sistema operativo, usadas por éste y por otros programas para guardar información necesaria para su funcionamiento.



ENLACE DE INTERÉS

Se puede ampliar información sobre las variables de entorno y sus valores en:

http://es.wikipedia.org/wiki/Variable_de_entorno

El ejemplo típico es la variable de entorno **PATH** con información acerca de la ubicación de varios directorios esenciales para el funcionamiento del ordenador.

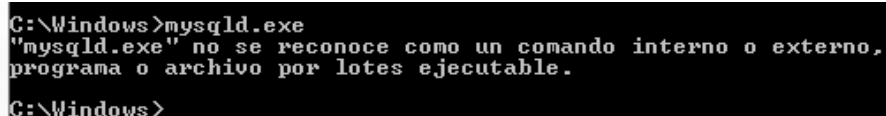


Imagen: ejemplo variables de entorno

5.1 EL PATH

Cuando se desea ejecutar un programa, sobre todo mediante línea de comandos pero también en otros casos, si no se ejecuta desde el mismo directorio donde está el ejecutable, es posible que el sistema operativo no sepa dónde encontrarlo, a él o a alguno de los componentes que necesite para funcionar, dando un mensaje de error.

Por ejemplo, si tenemos un servidor mysqld y queremos arrancarlo desde la línea de comandos, obtendríamos



```
C:\Windows>mysqld.exe
"mysqld.exe" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
C:\Windows>
```

Imagen: error al no encontrarse un comando

Para solucionarlo deberíamos o bien movernos hasta `c:\mysql\bin` (o donde quiera que esté el ejecutable) o bien desde el directorio que nos encontremos escribir el path íntegro, por ejemplo:

C:\WINDOWS>C:\mysql\bin\mysqld.exe

En cambio, si tuviéramos `c:\mysql\bin` en nuestro PATH, podríamos ejecutar cualquier programa situado dentro de bin desde cualquier directorio, sin movernos hasta allí ni escribir el path completo.

Por tanto, mediante el comando PATH se indican al sistema las **rutas de búsqueda** de programas ejecutables. De hecho, al instalarlos, muchos programas añaden automáticamente entradas en los ficheros de configuración indicando su propio path, lo que a su vez puede ser fuente de problemas, ya que un PATH indebidamente configurado puede dejar sin efecto el anterior, y porque el comando PATH tiene una importante limitación en cuanto al número máximo de valores que puede almacenar.

Para comprobar cuál es el valor actual de PATH en un sistema Windows basta con abrir una consola (mediante la ejecución del comando *cmd*). Tecleando PATH y pulsando Intro, veremos las rutas que la variable PATH tiene almacenadas.

```
C:\Windows>path
PATH=C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\ProgramData\Oracle\Java\javapath;C:\Program Files\Haskell\bin;C:\Program Files\Haskell Platform\2014.2.0.0\lib\extralibs\bin;C:\Program Files\Haskell Platform\2014.2.0.0\bin;C:\Program Files (x86)\ActiveState Komodo Edit 8\;C:\Python33\;C:\oraclexe\app\oracle\product\11.2.0\server\bin;C:\Program Files (x86)\HP SimplePass 2012\;C:\Pro
```

Imagen: contenido de PATH

Normalmente un PATH tendrá como mínimo la ruta `c:\windows` y `c:\windows\command`, lo que quiere decir que los programas que estén en esos directorios podrán ser ejecutados en línea de comandos desde cualquier otra carpeta del sistema, sin necesidad de movernos hasta la carpeta del ejecutable.

Un sistema Windows buscará el archivo a ejecutar siempre en primer lugar en el directorio actual, y sólo si allí no lo encuentra, sucesivamente en cada una de las rutas especificadas en PATH, de izquierda a derecha. Este dato es importante si vas a ejecutar un archivo del cual tienes varias copias con el mismo nombre.

PATH nos muestra el contenido de la variable PATH.

`PATH = ""` borra el contenido de la actual variable PATH; `PATH;` borra la propia variable PATH

`PATH = nueva ruta` o `SET PATH = nueva ruta` permite cambiar el contenido del valor PATH

`%PATH%` es una variable con el contenido de PATH

`PATH = %PATH%; nuevo_directorio` añade al PATH el nuevo valor sin borrar el anterior.

“

NOTA

- 1) Cada vez que se establece un nuevo PATH el anterior queda sin efecto, salvo que usemos la opción `%PATH%;nuevopath`
- 2) No podemos añadir al path nombres de directorios con espacios en blanco, como *Archivos de programa*. Para incorporarlos debemos entrar el valor entrecomillado o usar el nombre corto DOS: `Archiv~1`
- 3) Las distintas rutas se separan por ``;` y nunca por espacios en blanco

5.2 Variables de entorno en sistemas Windows

Para crear variables de entorno en Windows 10 (en versiones anteriores el proceso es muy similar), se debe acceder al Panel de control -> Sistema y seguridad -> Sistema

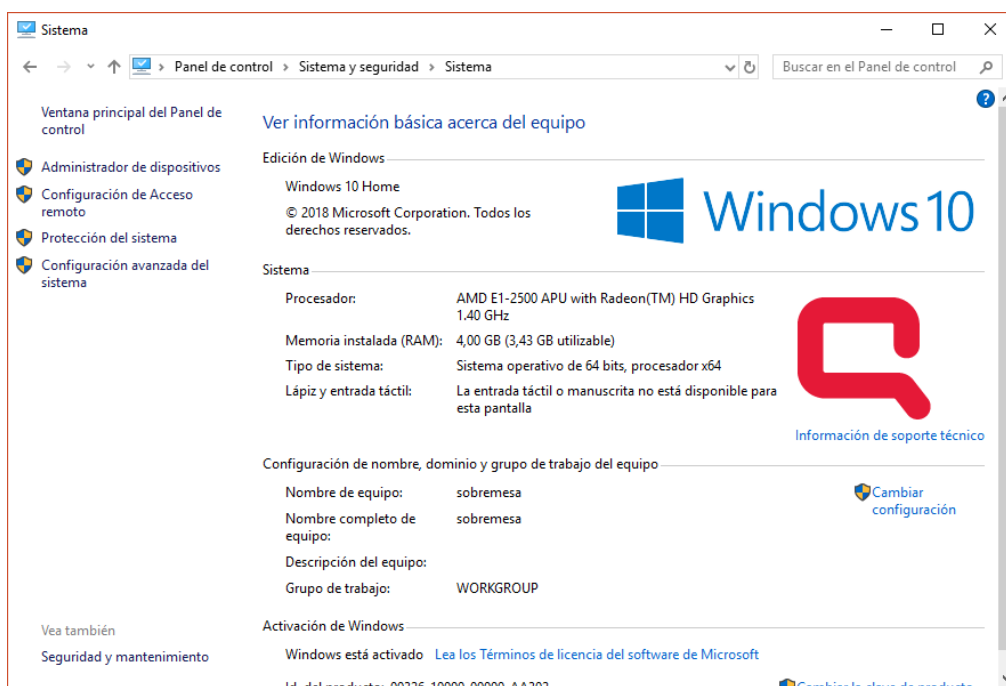


Imagen: acceso a las variables de entorno en Windows 10

Se elige Configuración avanzada del sistema, lo que nos mostrará la ventana de Propiedades del sistema

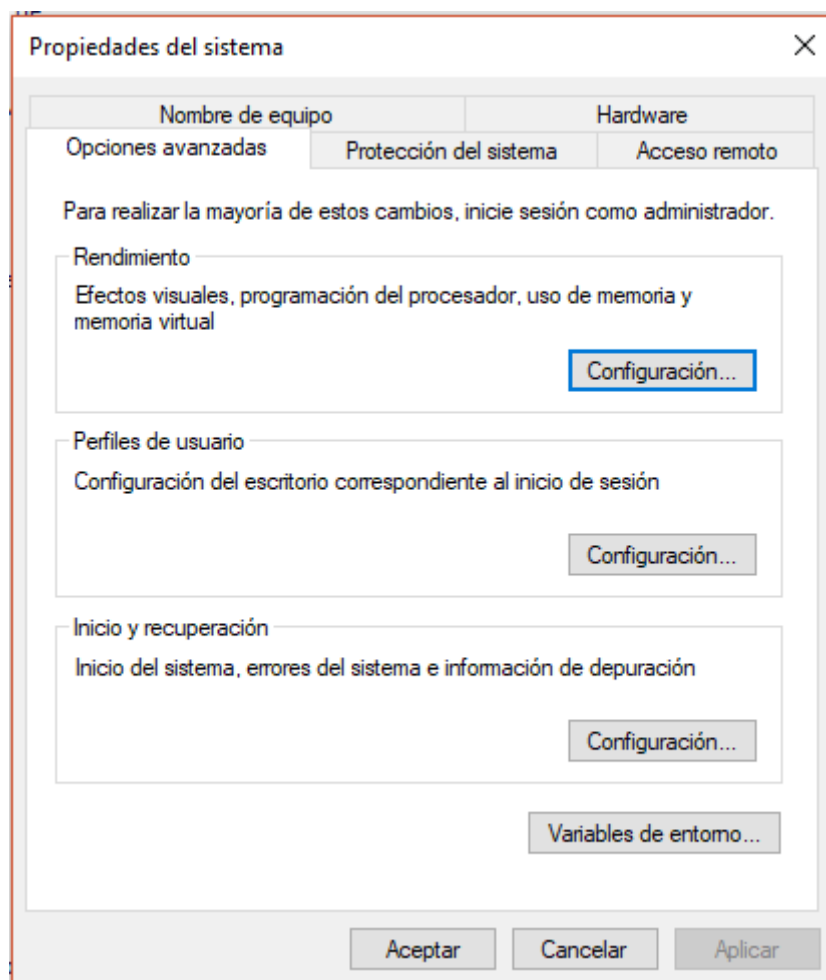


Imagen: propiedades del sistema

A continuación se pulsa el botón **Variables de entorno**, abriéndose una nueva ventana que nos muestra dichas variables

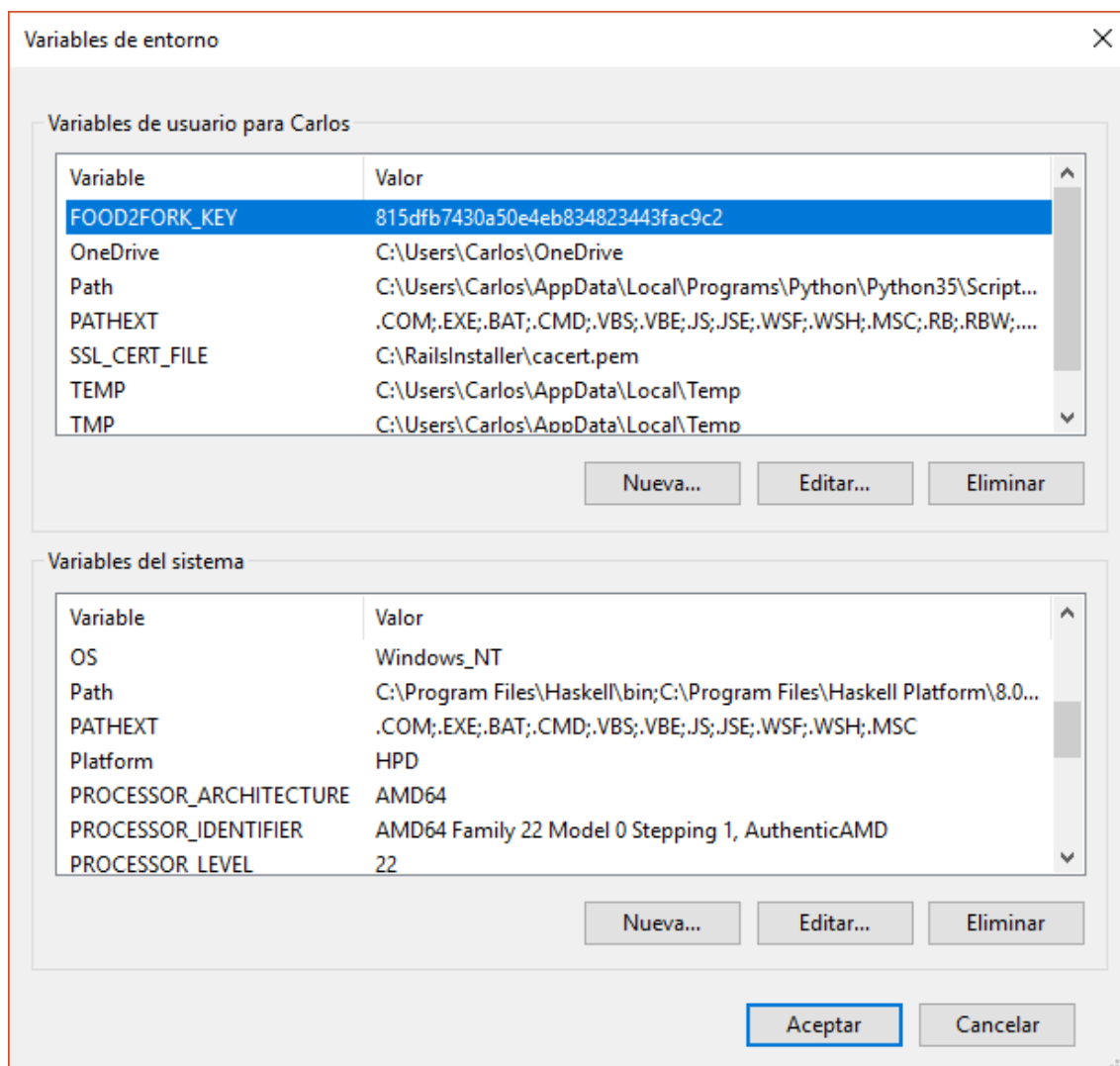


Imagen: variables de entorno

Estas variables de entorno pueden ser propias del usuario actual, o globales a todo el sistema. Si se desean editar, crear nuevas o eliminar alguna (esto siempre debe hacerse con precaución, pues el sistema podría dejar de funcionar correctamente), se puede hacer uso de los correspondientes botones (Nuevo, Modificar, Eliminar).

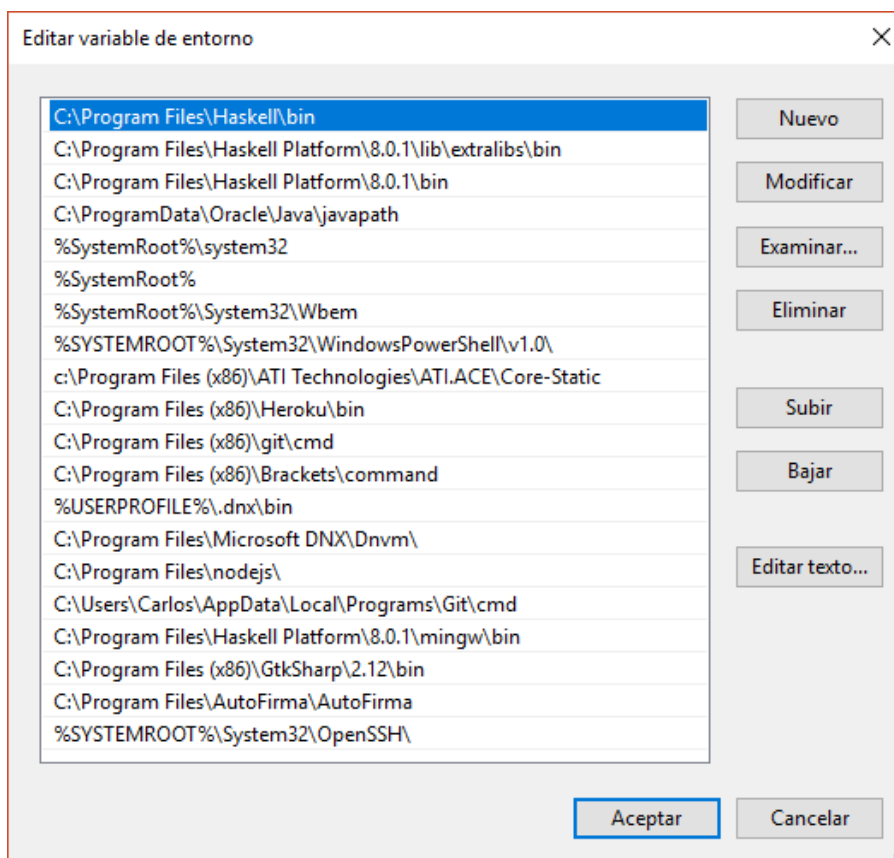


Imagen: edición de variables de entorno

5.3 Variables de entorno en sistemas Linux

Las **variables de entorno**, como se ha visto para Windows, pueden ser diferentes para cada usuario, y permiten personalizar el entorno Linux a gusto del usuario.

Por lo general, en UNIX las variables de entorno se escriben en mayúsculas (no hay nada que impida que vayan en minúsculas, salvo evitar confusiones con comandos). Las llamadas a las variables de entorno se realizan anteponiendo el signo "\$" al nombre de la variable (\$HOME, \$PS1, etc.).

Las variables de entorno no pueden contener el carácter "\$" ni espacios, salvo en el valor, siempre y cuando este se presente entrecomillado.

En UNIX existen unas variables de entorno básicas, que son necesarias para que el usuario pueda trabajar adecuadamente. Además, tanto el propio usuario como algunos programas que este ejecute, pueden generar nuevas

variables de entorno. Se puede así distinguir entre variables de entorno generales y variables de entorno particulares.

Las variables de entorno generales difieren poco, en cuanto a nombre se refiere, entre los distintos sistemas UNIX, y expresan lo mismo en cada sistema.

Las variables de entorno se pueden definir en los archivos de inicialización, al arrancar un programa, o desde la shell de usuario, y en cualquiera de los casos se definen de la siguiente forma:

NOMBRE=valor

Si se desea exportar una variable, para que sea visible fuera del programa, habrá que hacer

export NOMBRE

o bien

export NOMBRE=valor

Para visualizar el valor de una variable de entorno basta con teclear:

echo \$NOMBRE

Entre las variables generales más habituales encontramos:

EDITOR: Variable que almacena el editor que usará el usuario por defecto.

ENV: Variable que almacena el archivo en donde el usuario ha definido sus alias y funciones.

HOME: Variable que almacena el directorio del usuario, desde el que arrancará la shell cuando entra en el sistema.

HOSTNAME: Variable que almacena el nombre de la máquina.

LOGNAME: Variable que almacena el nombre o login del usuario.

MAIL: Variable que almacena el archivo que contiene el correo de usuario.

MAILCHECK: Variable que indica cada cuantos segundos debe comprobarse si el usuario tiene correo.

PATH : Variable en la que se encuentran almacenados los paths de aquellos directorios a los que el usuario tiene acceso directo, pudiendo ejecutar comandos o programas ubicados en ellos sin necesidad de acceder a dicho directorio.

PS1: Variable que almacena el prompt que se empleará por defecto en la shell.

PS2, PS3, etc.: Variable que almacena el prompt que se empleara en la 2ª, 3ª, etc.

Subshells, lo cual se hace invocando a la shell, es decir, ejecutando por ejemplo, ksh.

PWD: Variable que almacena el directorio actual, puede ser útil para modificar el prompt (PS1) dinámicamente.

SHELL: Variable que almacena el intérprete de comandos que usa el usuario.

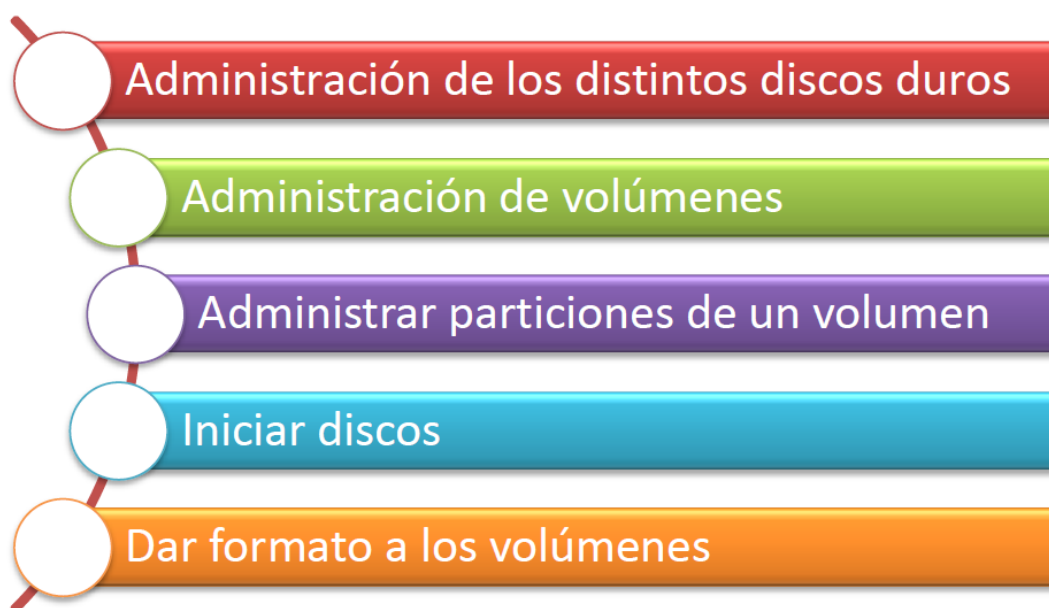
TERM: Variable que almacena el tipo de terminal desde el que se está trabajando.

TMOUT: Variable que almacena el tiempo de inactividad que se permite al usuario antes de que el sistema le cierre la sesión.

6. ADMINISTRACIÓN DE DISCOS

En este apartado se mostrarán los ejemplos principalmente en sistemas Windows. Cabe indicar que los sistemas Linux tienen herramientas muy similares a las que se van a describir aquí.

En los sistemas operativos Windows existe una herramienta llamada "**Administrador de discos**" que facilita las siguientes operaciones:



Para acceder al administrador de disco tenemos varias opciones

- Inicio → Panel de control → Herramientas administrativas → Administración de equipos → Almacenamiento → Administración de discos
- Inicio → Ejecutar → **diskmgmt.msc**

Al acceder al administrador de discos de algunas de las maneras explicadas, nos aparece la siguiente ventana, donde se aprecian los dispositivos de almacenamiento del equipo, así como el sistema de archivos de cada partición, tipo de unidad y partición, entre otros aspectos.

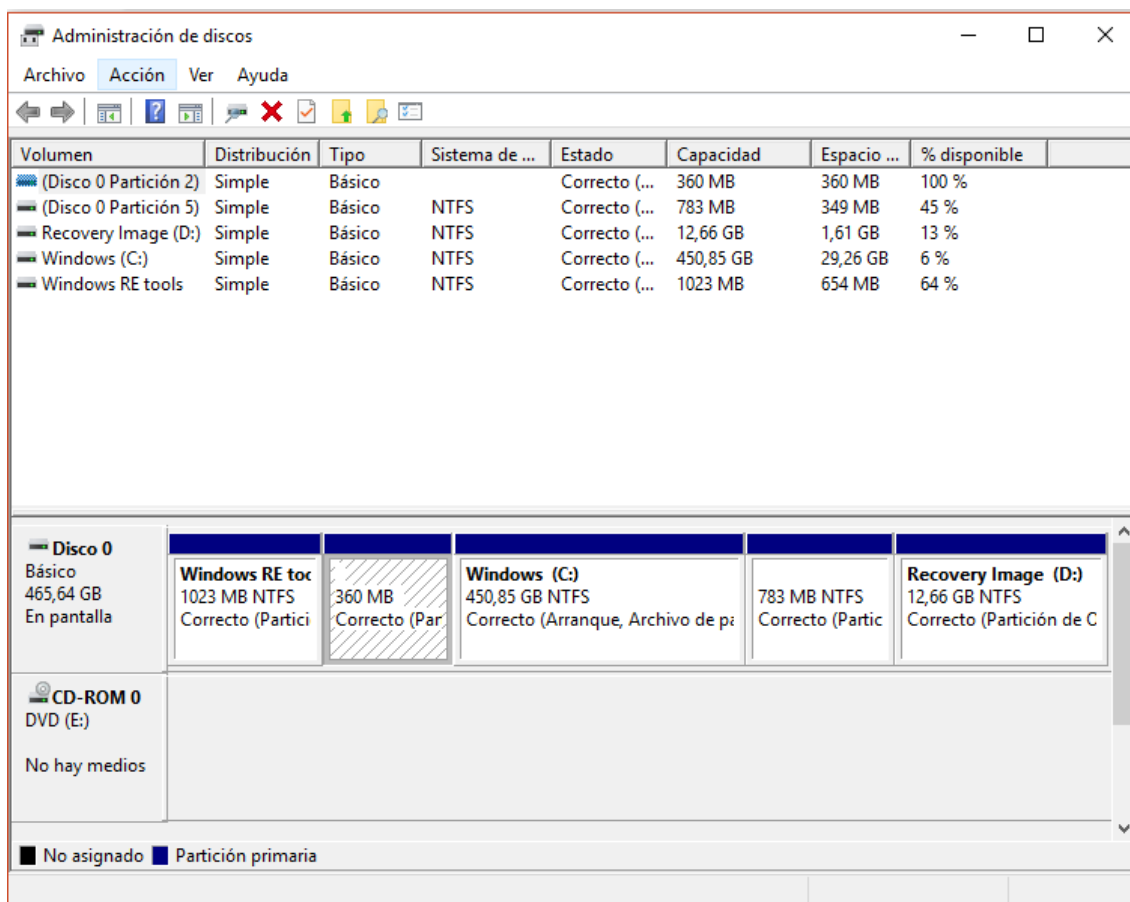


Imagen: administrador de discos en Windows

6.1 Particiones, unidades lógicas y volúmenes

A la división de un disco duro en trozos más pequeños se le denomina **partición**. Gracias a ellas podemos dividir o trocear un disco duro en diferentes zonas donde instalar otros sistemas operativos o almacenar datos, almacenar otro tipo de información etc.

Normalmente, al comprar un ordenador con un sistema operativo ya instalado, éste viene instalado en un único disco duro sin particionar. Pero como las unidades de disco aumentan cada vez más su capacidad, llegados a este punto cabe preguntarse si es conveniente tener todo ese espacio formateado junto. La respuesta es que no, y es bastante recomendable establecer un sistema de particiones que nos permita estructurar nuestra información de manera óptima. Un ejemplo muy típico sería separar los datos de los usuarios en una partición independiente. O emplear una partición para realizar copias de seguridad.

La idea es que tener varias particiones es como disponer de diferentes discos duros en un solo disco físico, cada uno de ellos con su propio sistema de archivos y funcionando de manera independiente.

Hay dos tipos de particiones:



Las **particiones primarias** son las que se detectan al arrancar el ordenador, por lo que habitualmente se instalan en ellas los sistemas operativos.

Las **particiones extendidas** o secundarias se crearon para superar el límite máximo de 4 particiones que tradicionalmente se establecía en los sistemas operativos. Normalmente se emplean para almacenar datos, y suele haber una sola de ellas en cada equipo. Eso sí, se puede dividir en tantas **unidades lógicas** como se desee, que igualmente actúan como discos duros independientes.

Otro concepto relacionado es el de **volumen**, que a veces se emplea como sinónimo de partición. En realidad no son idénticos: la partición es la división física de un disco duro, mientras el volumen es su representación lógica, una vez el sistema operativo ya ha preparado dicha partición para su uso.

6.1.1 Creación de particiones en Windows 10

En primer lugar, se accede a la herramienta de Administración de discos mediante la opción del Panel de control "Crear y formatear particiones del disco duro".

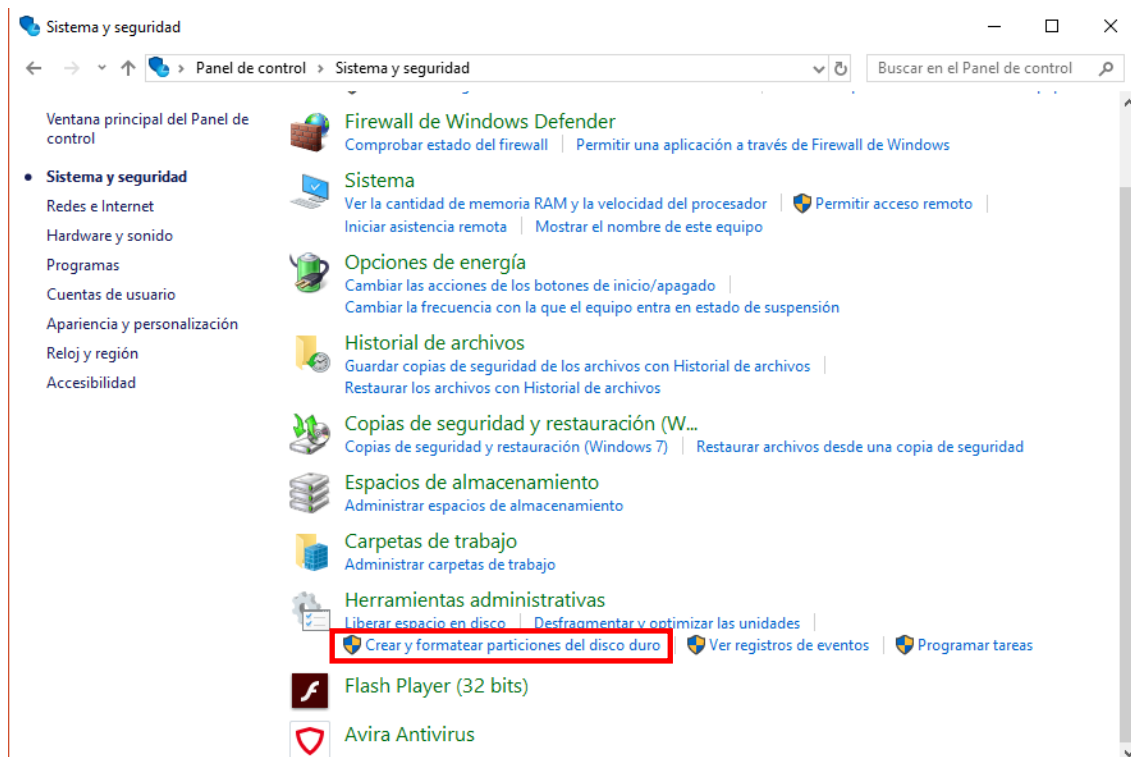


Imagen: acceso al administrador de discos en Windows 10

Ya en el administrador de discos, para empezar hay que reducir el tamaño de la partición principal, generando así espacio suficiente para albergar la nueva. Para ello, se pulsa sobre dicha partición con el botón derecho del ratón, y se elige "Reducir volumen".

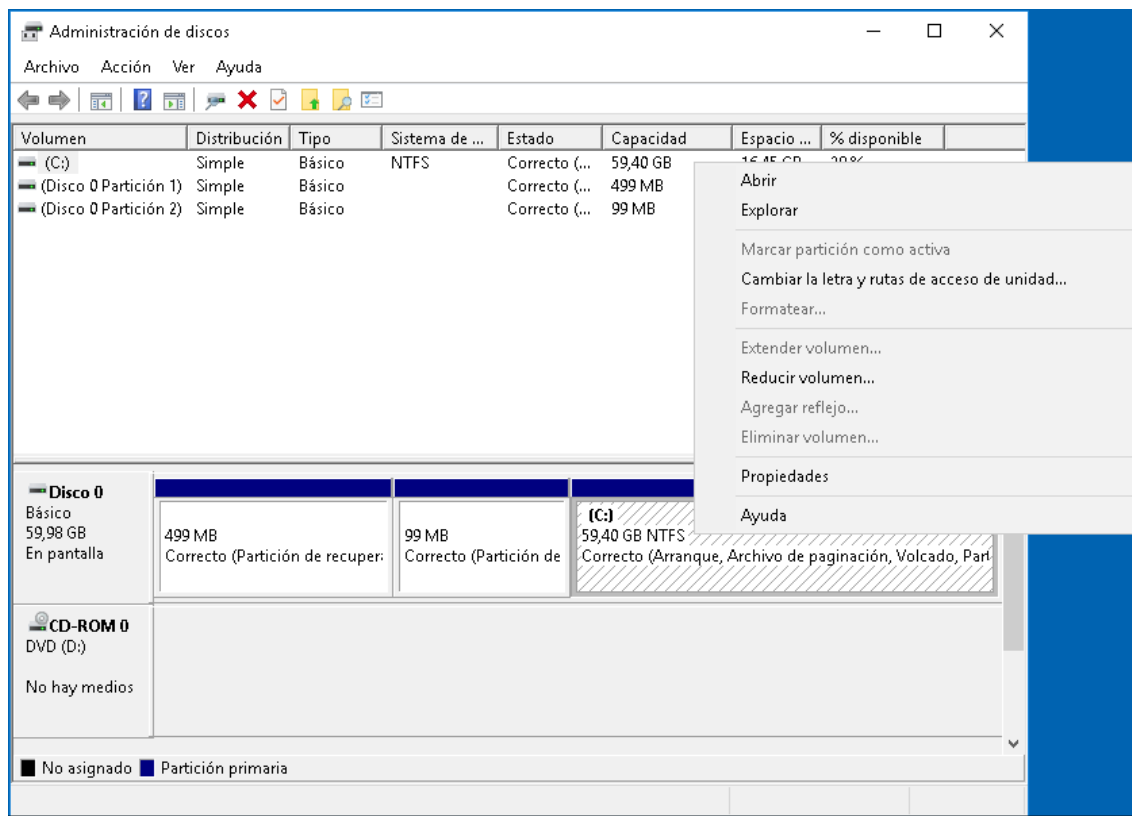


Imagen: generar espacio para la nueva partición

Se van a liberar 10 GB para la partición nueva.

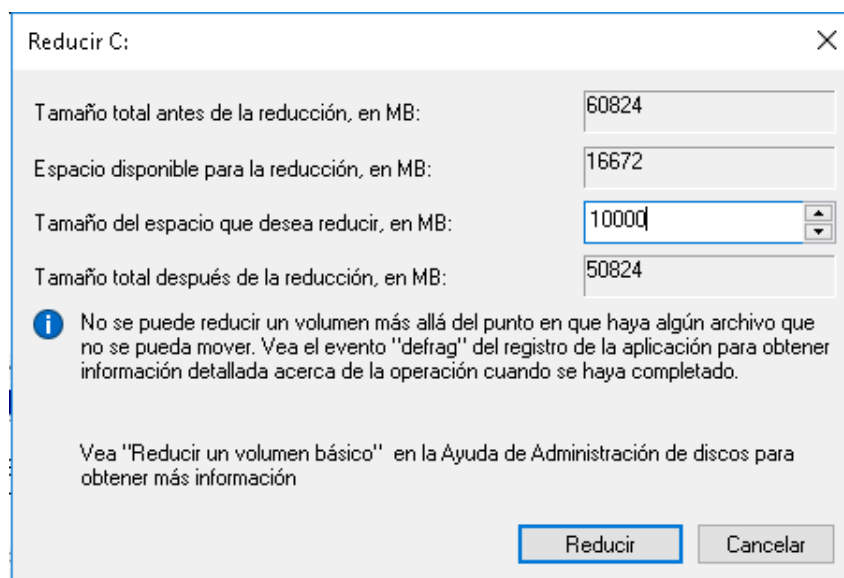


Imagen: reducción en 10 GB de la partición inicial

El espacio que acabamos de liberar aparece ahora como "No asignado". Para crear sobre él la nueva partición, se pulsa con el botón derecho y se elige **"Nuevo volumen simple"**

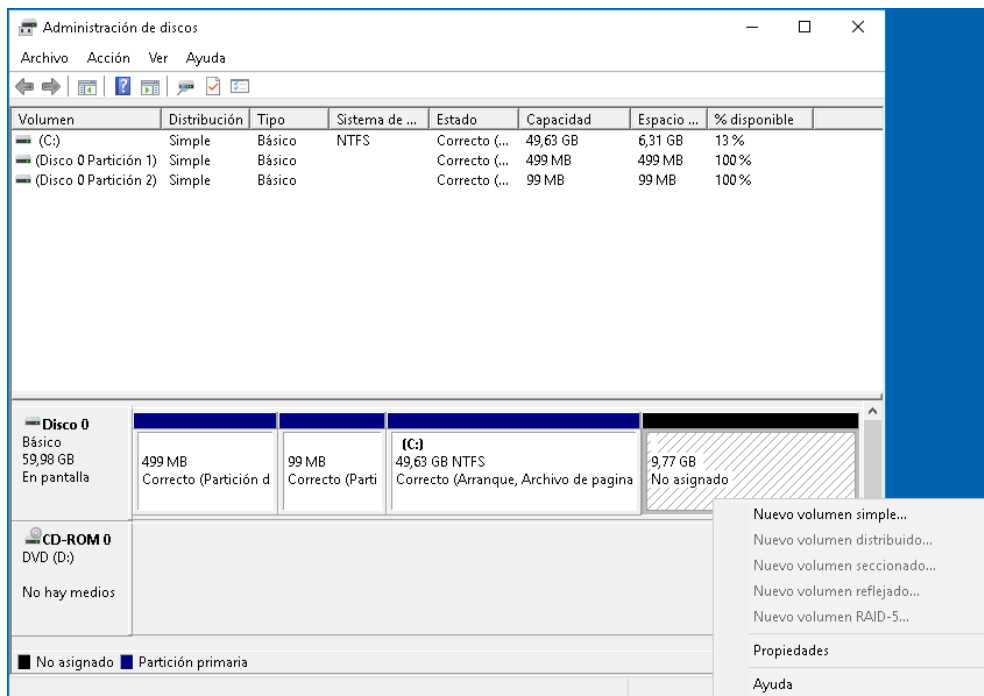


Imagen: creación de la nueva partición

Un asistente guía el proceso, en el que se pedirá en primer lugar especificar el tamaño de la partición. Se asignan los 10 GB disponibles, ya que en este caso sólo se desea crear una nueva partición.

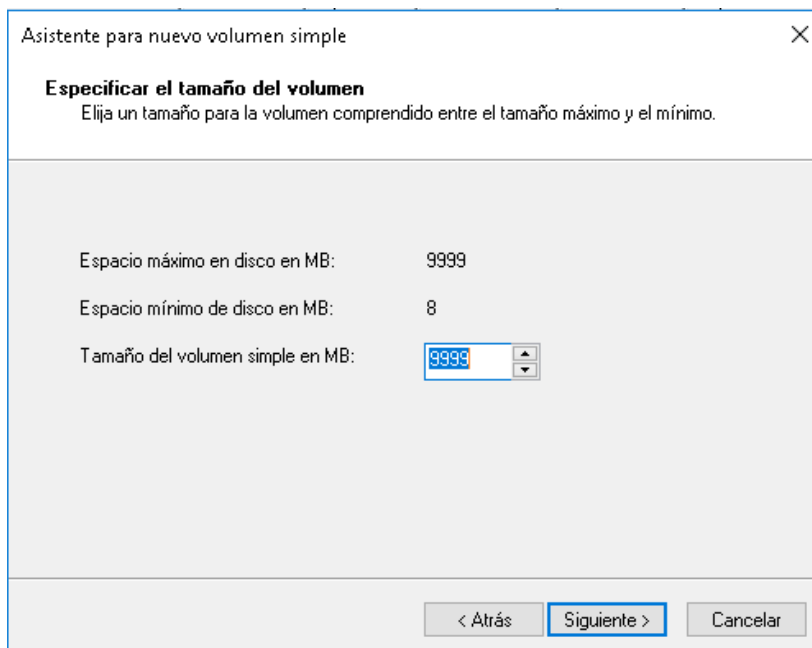


Imagen: especificación del tamaño de la nueva partición

Seguidamente se solicita una letra de unidad para la nueva partición. Se asigna la siguiente disponible, en este caso la E:

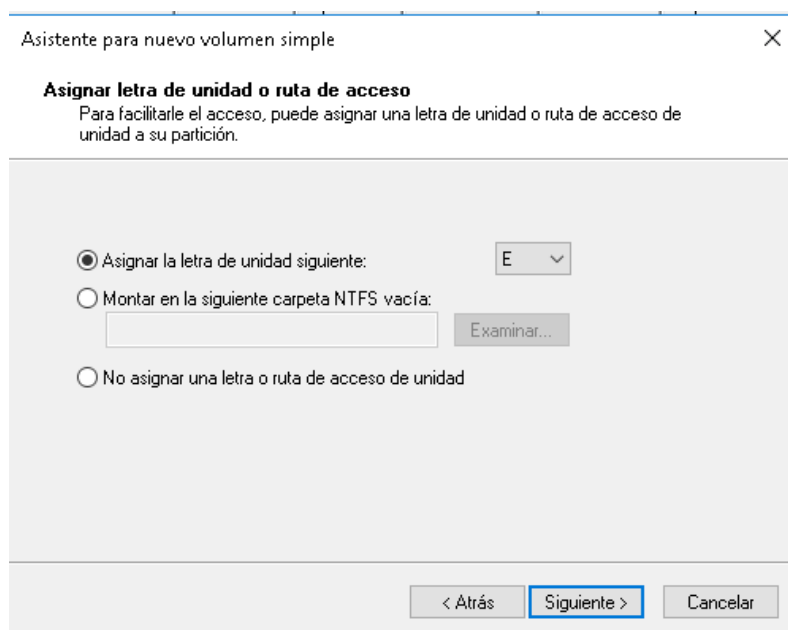


Imagen: asignación de letra de unidad

Por último se ofrece la opción de formatear el nuevo volumen creado, seleccionando el sistema de archivos deseado

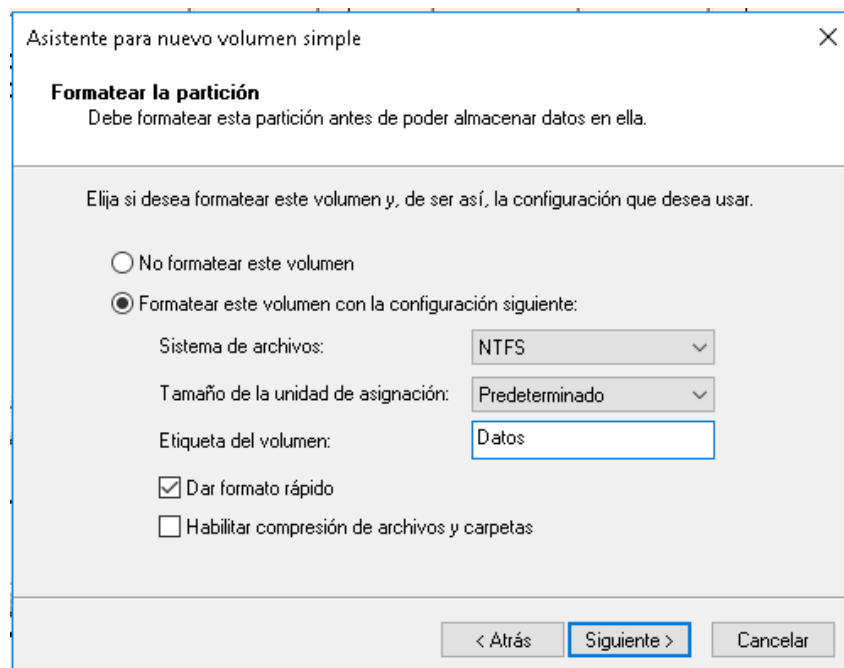


Imagen: formateo de la nueva partición

Se finaliza el asistente para que se apliquen todas las opciones elegidas

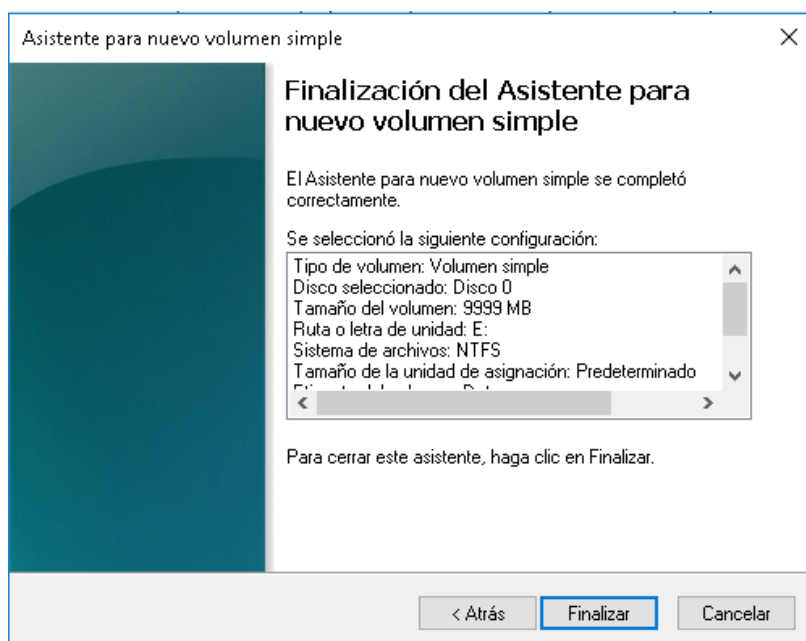


Imagen: finalización del asistente

Y ya está creada la nueva partición.

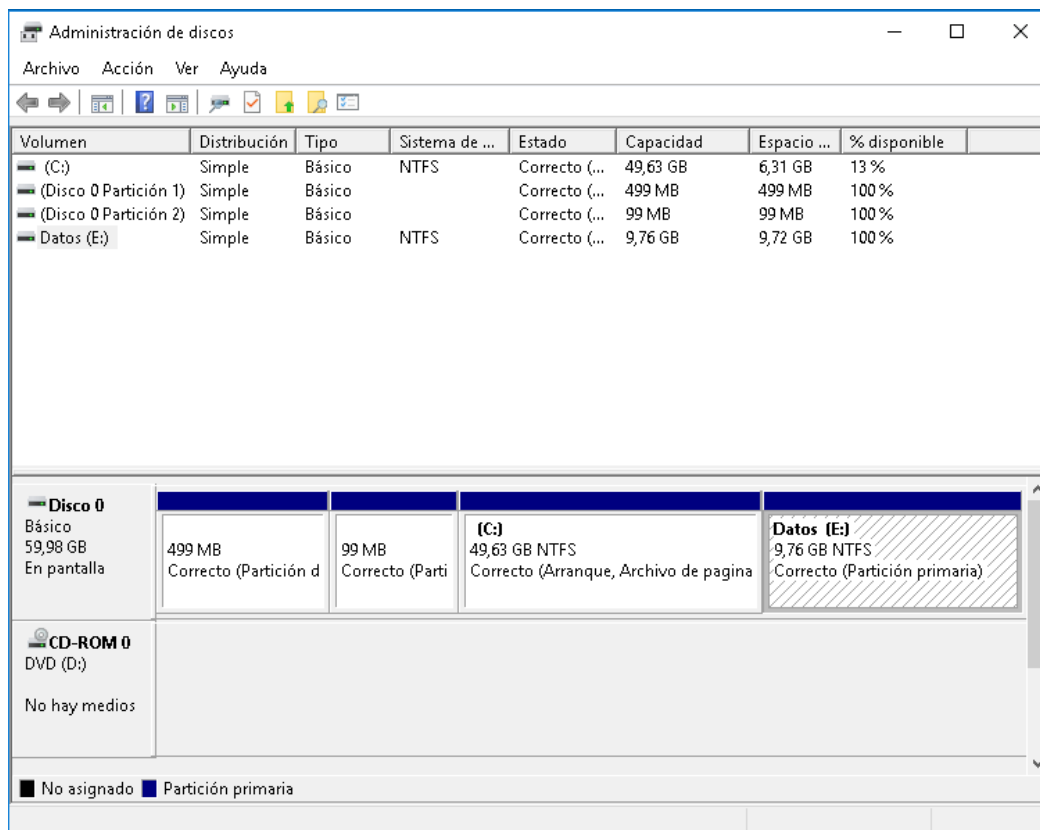


Imagen: partición nueva creada

6.1.2. Particiones en Linux

Respecto a los sistemas Linux, aunque como se comentaba con anterioridad los conceptos y herramientas son muy similares, citar que se requieren al menos 2 particiones:

- 1.- Una donde se guardarán todos los datos.
- 2.- Una partición especial denominada **swap** (partición de intercambio) utilizada como memoria virtual.




Partición		Sistema de archivos	Punto de montaje	Tamaño	Usado	Libre
/dev/sda1		ext4	/	18.48 GiB	4.79 GiB	13.70 GiB
/dev/sda2		extended		1.51 GiB	---	---
/dev/sda5		linux-swaps		1.51 GiB	---	---

Imagen: particiones en Linux

La partición swap es usada internamente por el sistema, y por lo tanto, el usuario no puede utilizarla para almacenar archivos o crear directorios. El sistema utiliza la partición swap como si fuera memoria cuando se queda sin memoria física. La función que cumple la partición swap también se puede llevar a cabo mediante un archivo (como ocurre en los sistemas Windows, lo que se conoce como archivo de paginación), aunque esta opción no se suele utilizar por cuestiones de rendimiento.

Otra característica que se debe conocer es la forma en la que Linux **nombra los discos y particiones**. Hasta la versión del kernel 2.6.18, los discos duros de tipo IDE se nombraban como /dev/hd seguido de una letra que identificaba la unidad, por ejemplo: /dev/hda. Mientras que los discos duros de tipo SCSI y los discos externos o unidades flash USB se identificaban como /dev/sd seguido de una letra que identificaba la unidad, por ejemplo: /dev/sda.

A partir de la versión del kernel 2.6.19, donde además se incluye soporte para discos Serial ATA, el prefijo utilizado para nombrar tanto discos duros como unidades externas conectadas por USB es /dev/sd.

Así por ejemplo, en un equipo con 2 discos duros, éstos se nombrarán como /dev/sda y /dev/sdb.

Si conectamos un disco externo USB normalmente se le asignará el nombre /dev/sdc.

Este formato de nombres también se utiliza para nombrar las particiones de los discos. Para ello se añade al nombre del disco un número que identifica la partición dentro de la unidad. Así, la primera partición del primer disco duro del sistema se nombra como /dev/sda1.

6.2 Desfragmentación

La **desfragmentación** de un disco consiste en concentrar los ficheros y directorios fragmentados, para que cada uno ocupe un solo espacio contiguo en el volumen. Es usual que cuando trabajamos con documentos grandes, los cuales se actualizan cada cierto tiempo (bases de datos, documentos de texto enriquecido, etc.), dichos archivos terminen fragmentándose en distintos trozos de manera no consecutiva por el disco duro.

En Windows 10 se realiza desde el Panel de Control, empleando la opción **“Desfragmentar y optimizar las unidades”**

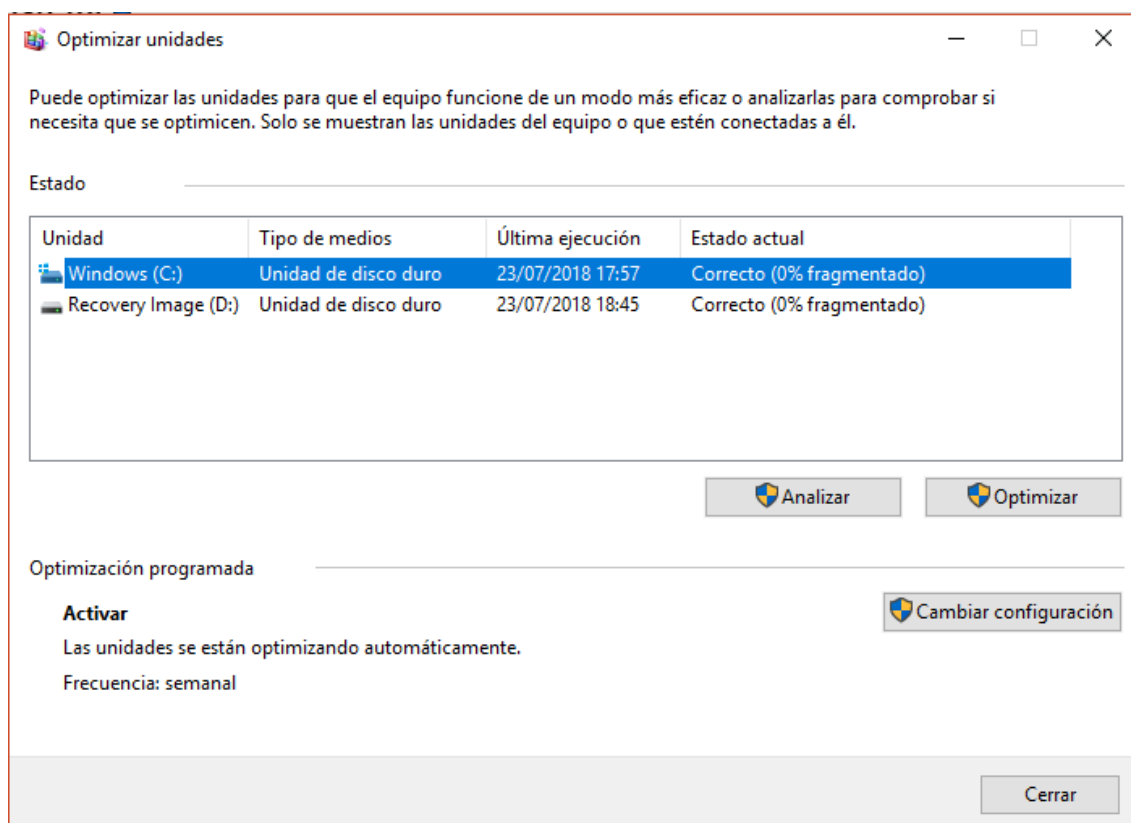


Imagen: herramienta de Desfragmentación en Windows 10

Esta aplicación nos informa directamente de si nuestras unidades precisan o no desfragmentarse. En cualquier caso, se dispone de botones para Analizar y Optimizar la unidad deseada.

6.3. Copias de seguridad

Es frecuente que la información guardada en dispositivos de almacenamiento se pierda debido a causas naturales, borrado intencionado o accidental, fallo mecánico u otros motivos. Gracias a las copias de seguridad se podrán recuperar los datos perdidos.

Antes de realizar una copia de seguridad, conocida habitualmente por su término en inglés, *backup*, hay que tener en cuenta los siguientes aspectos:



Imagen: Factores a tener en cuenta para realizar copias de seguridad

6.3.1 Tipos de copias de seguridad

Se pueden dividir las copias de seguridad en tres categorías distintas, dependiendo de las necesidades, teniendo en cuenta que cada tipo de backup funciona de forma diferente con sus ventajas e inconvenientes:



Imagen: Tipos de copias de seguridad

Copia de seguridad completa, total o íntegra

Se copian la totalidad de los datos (archivos o carpetas) en la ruta donde se indique (por lo general otro soporte de almacenamiento).

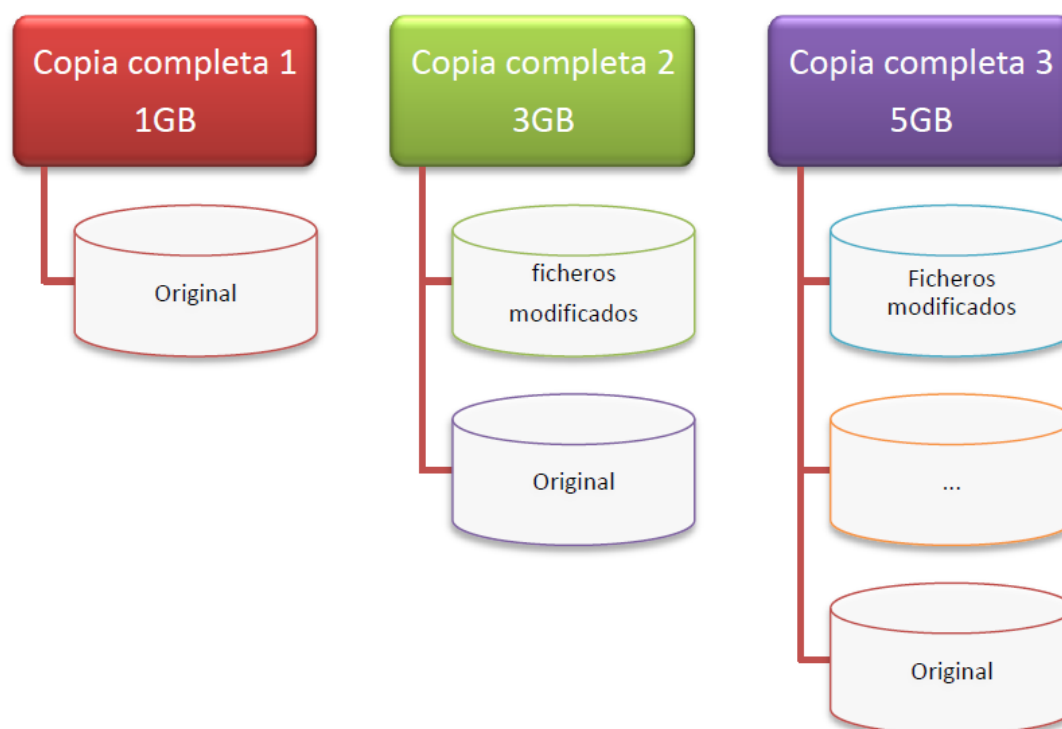


Imagen: copia de seguridad completa

Ventajas:

- Se dispone de la totalidad de los datos en un único archivo.
- Esto permite restaurar los datos en un tiempo mínimo

Inconvenientes:

- Lleva más tiempo realizar un backup completo que de otro tipo y requiere más espacio de almacenamiento

Copia de seguridad diferencial

Se copiarán todos los datos que se hayan modificado **desde la última copia completa** que se hiciera en el sistema. Por lo tanto, se almacenarán todos los archivos que se hayan modificado y los nuevos, siempre con respecto al último backup completo. Respecto a la incremental, se copian más ficheros, ocupa más espacio y suele tardar más tiempo. La ejecución de un backup diferencial requiere menos espacio y tiempo que la de los backup completos.

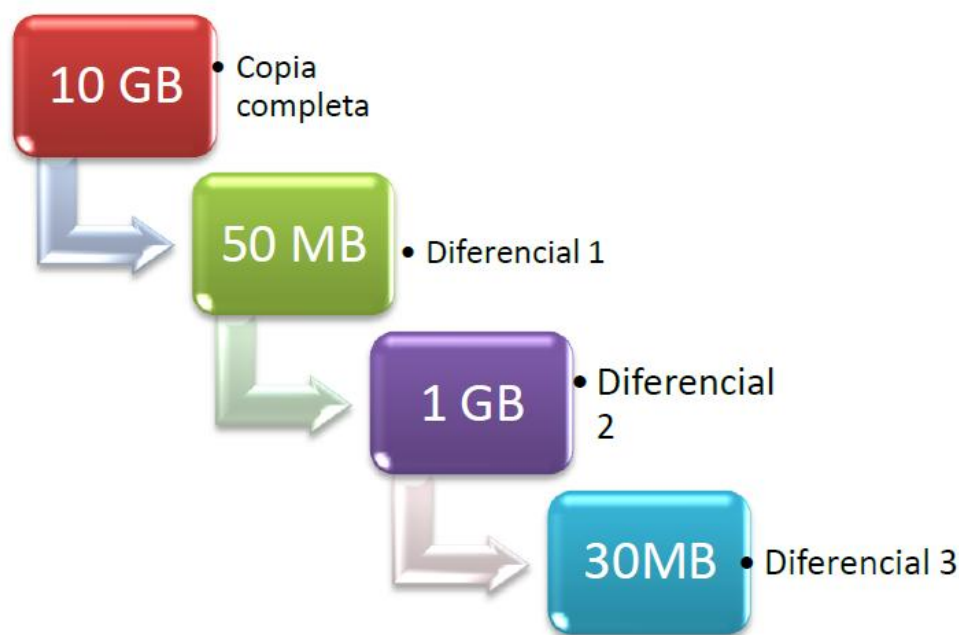


Imagen: copia de seguridad diferencial

Copia de seguridad incremental

Es frecuente que un usuario realice a lo largo del día operaciones sobre unos pocos de ficheros, ya sea creación, borrado o modificación de documentos. Este tipo de copia de seguridad **sólo** copia los datos que han variado, incluyendo los nuevos, desde la última operación de backup.

Se suele utilizar la hora y fecha de modificación de los archivos (atributo modificado), comparándola con la hora y fecha del último backup. **Es la que menos espacio ocupa.**

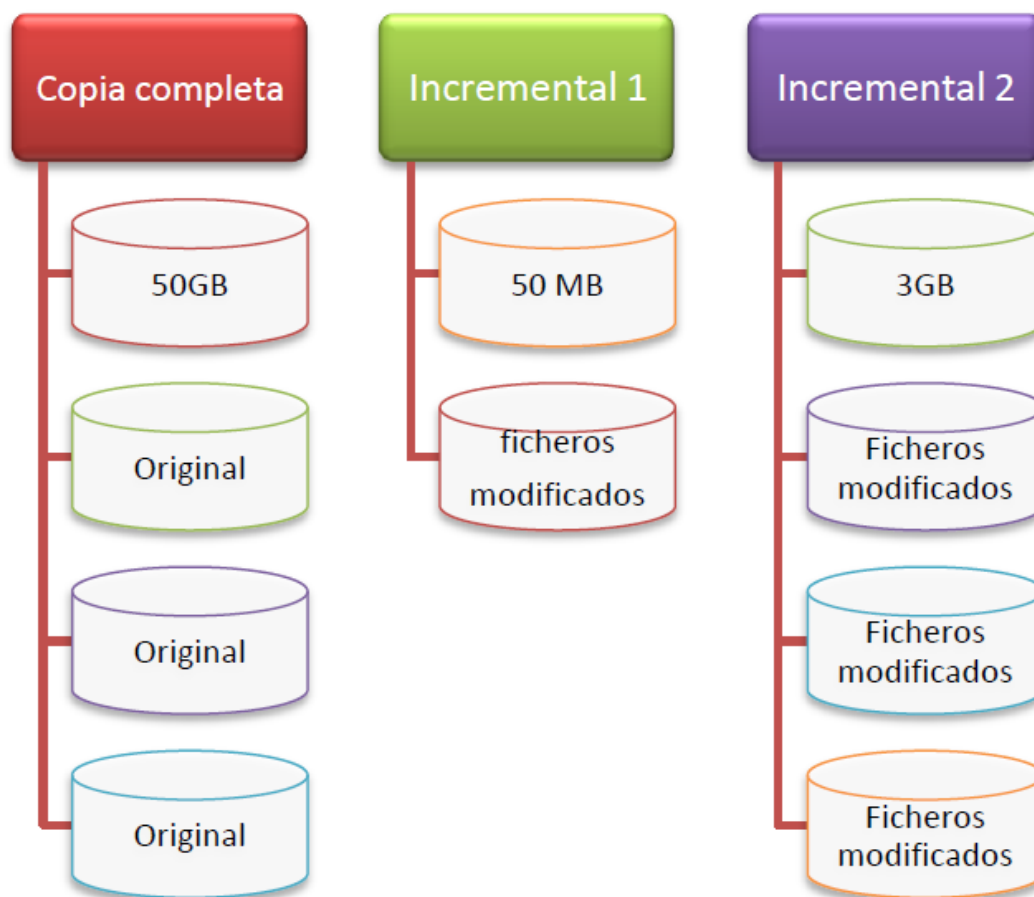


Imagen: copia de seguridad incremental

6.3.2 Creación de copias de seguridad

Todo sistema operativo proporciona herramientas para la realización de copias de seguridad. Por ejemplo, en Windows 10 se puede acceder a dicha herramienta de gestión de las copias de seguridad desde el Panel de control.

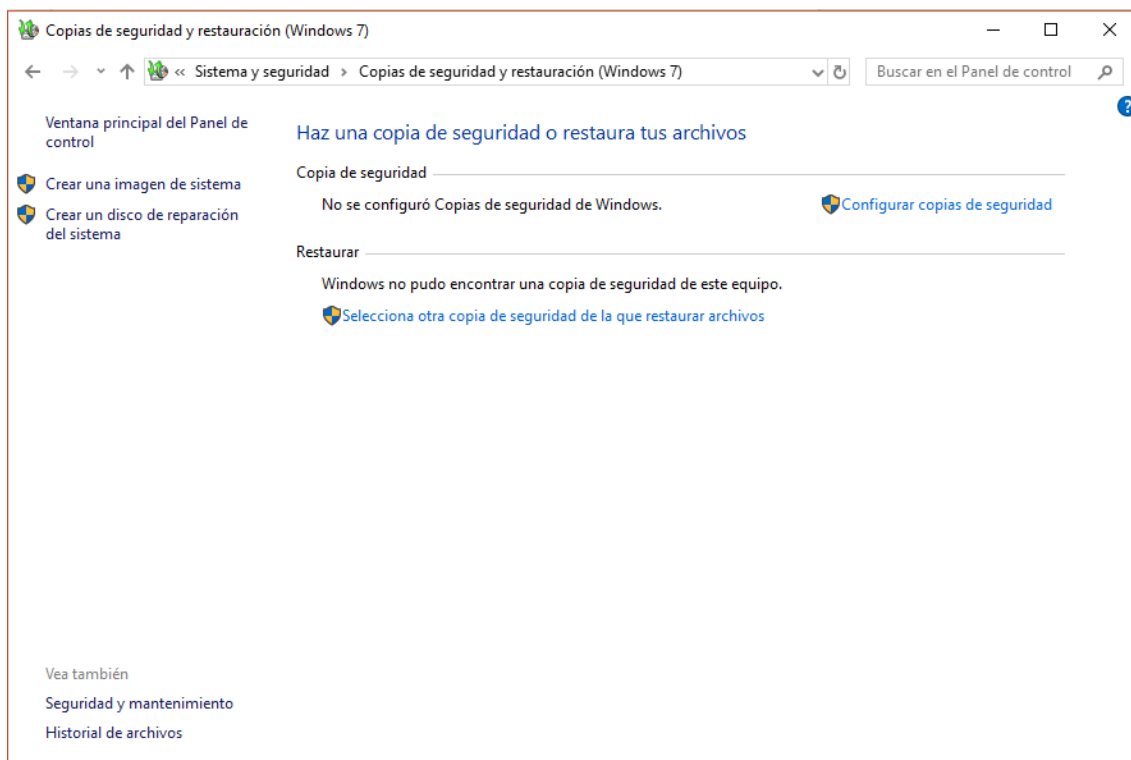


Imagen: acceso a la herramienta de copias de seguridad en Windows 10

Mediante “**Configurar copias de seguridad**” se puede crear una nueva copia, eligiendo el lugar en que se desea almacenarla

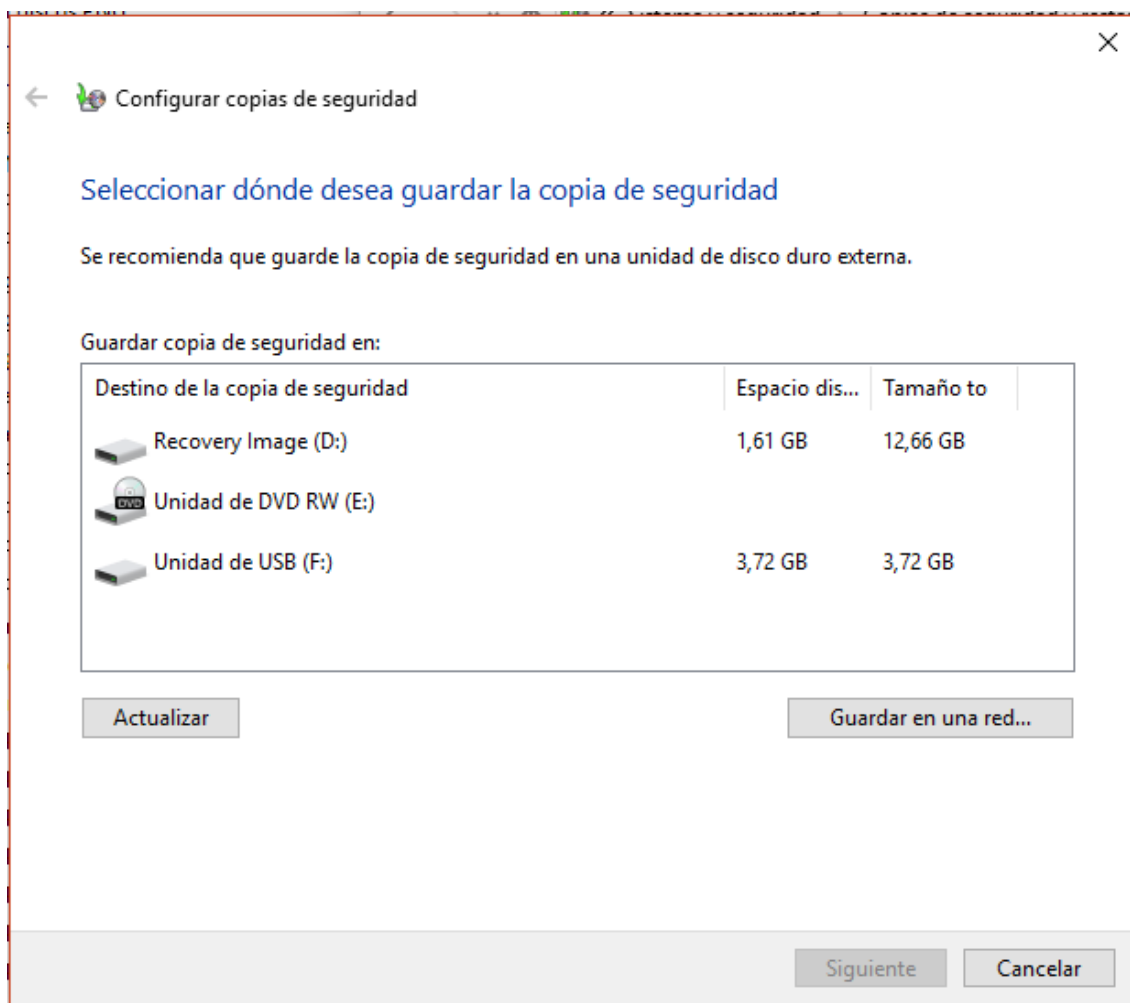


Imagen: configuración de la nueva copia de seguridad

Una vez creada la copia, podrá recuperarse en caso de necesidad mediante las opciones de Restaurar de la ventana mostrada anteriormente.

Gracias a los **planes de copia de seguridad**, se pueden programar por parte del administrador copias de seguridad de manera automática, especificando una serie de características adecuadas a la empresa.

6.4. RAID

Por regla general, en un sistema doméstico basta con adoptar las medidas básicas de administración de discos que se han descrito en esta unidad: **particionado, desfragmentación, copias de seguridad**. Pero en sistemas más complejos se añaden necesidades adicionales, en cuanto a capacidad, integridad, tolerancia a fallos, alta disponibilidad...

Es por eso que aparecen técnicas más avanzadas, que simplemente se comentarán a modo introductorio. Como por ejemplo los sistemas de

almacenamiento **RAID** (Redundant Array of Independent Disks), que utilizan múltiples discos entre los cuales se distribuyen o replican los datos.

Hay diferentes niveles de RAID, que proporcionan distintas prestaciones para satisfacer los requerimientos concretos en cada caso. El nivel más simple, RAID 0, combina varios discos duros en una sola unidad lógica. El ejemplo que se va a mostrar es el del siguiente nivel, RAID 1, que permite reflejar dos discos de manera que ambos contengan una copia exacta de sus datos. Esto aumenta la seguridad de nuestros datos, ya que si un disco falla tenemos otro con idéntica información. Aunque a cambio, lógicamente, perderemos capacidad de almacenamiento.

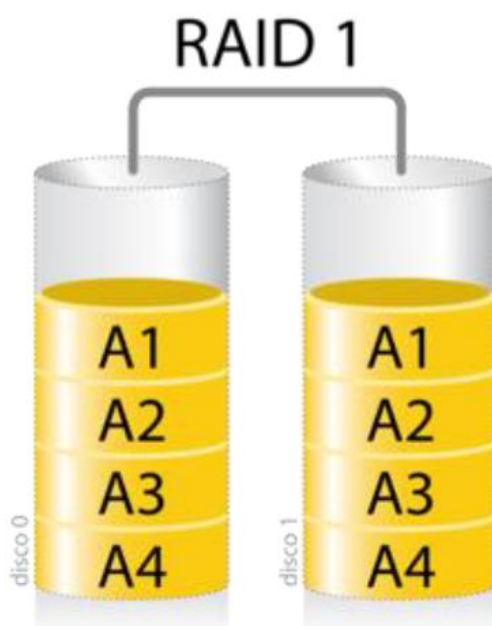


Imagen: RAID 1

Para implementar RAID 1 en Windows 10, en primer lugar necesitamos disponer de dos discos del mismo tamaño, que se convertirán en reflejo el uno del otro. Si estamos usando virtualización, basta con añadir dos nuevos discos a nuestra máquina virtual. En este caso se han añadido dos discos de 5 GB. Accedemos al Administrador de discos.

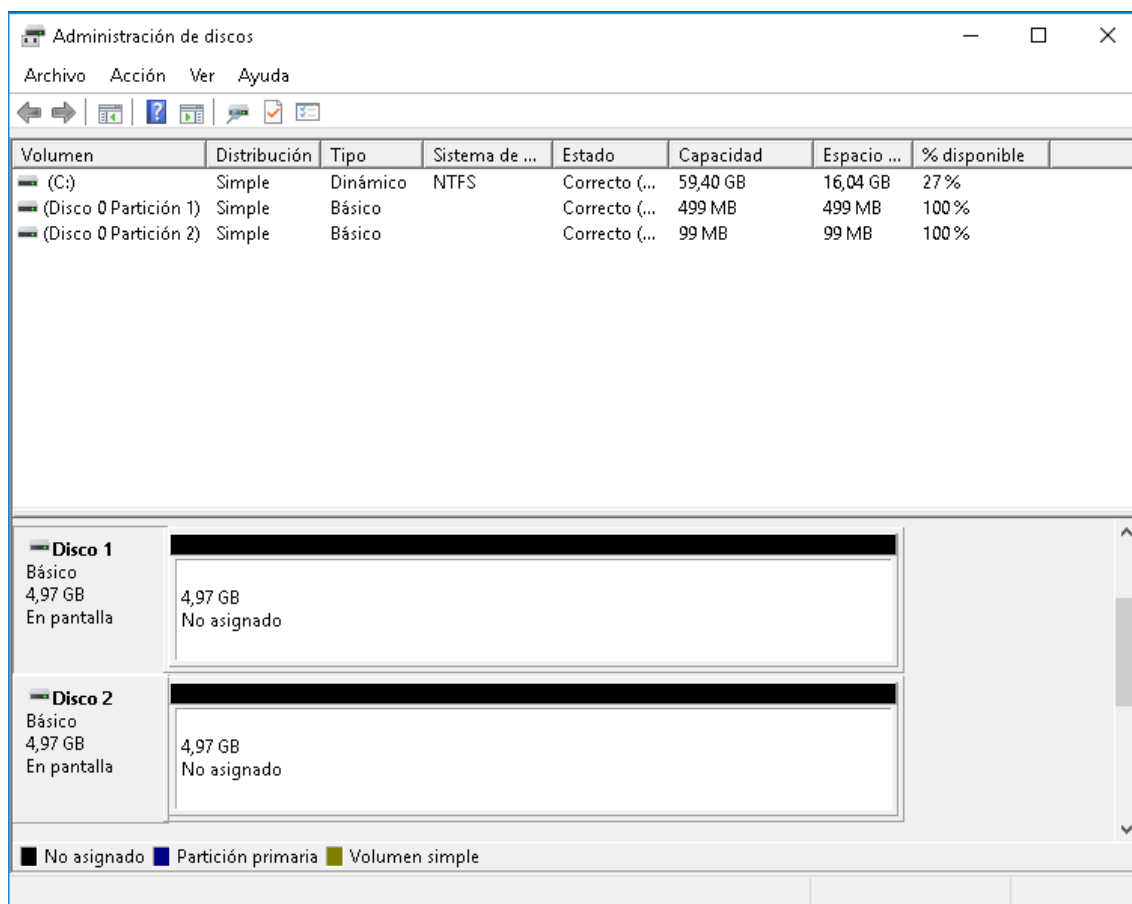


Imagen: Implementación de RAID 1 mediante el administrador de discos

Se pulsa con el botón derecho sobre el primero de los nuevos discos, y se elige **"Nuevo volumen reflejado"**

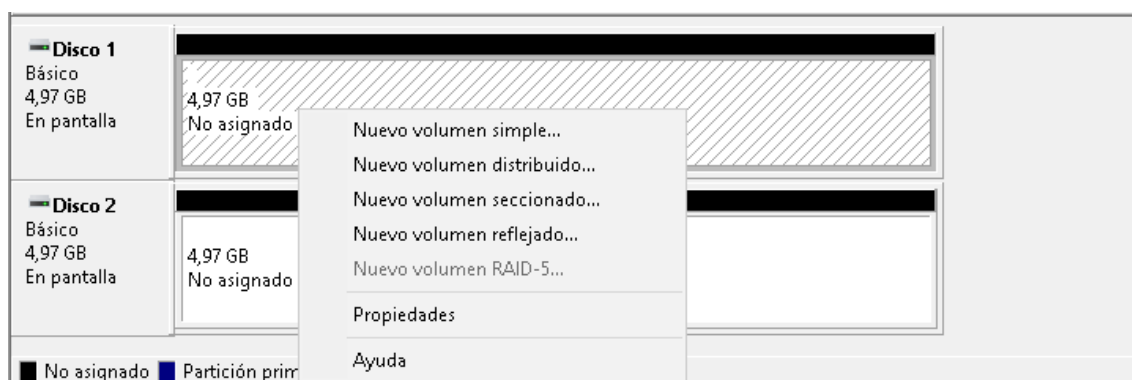
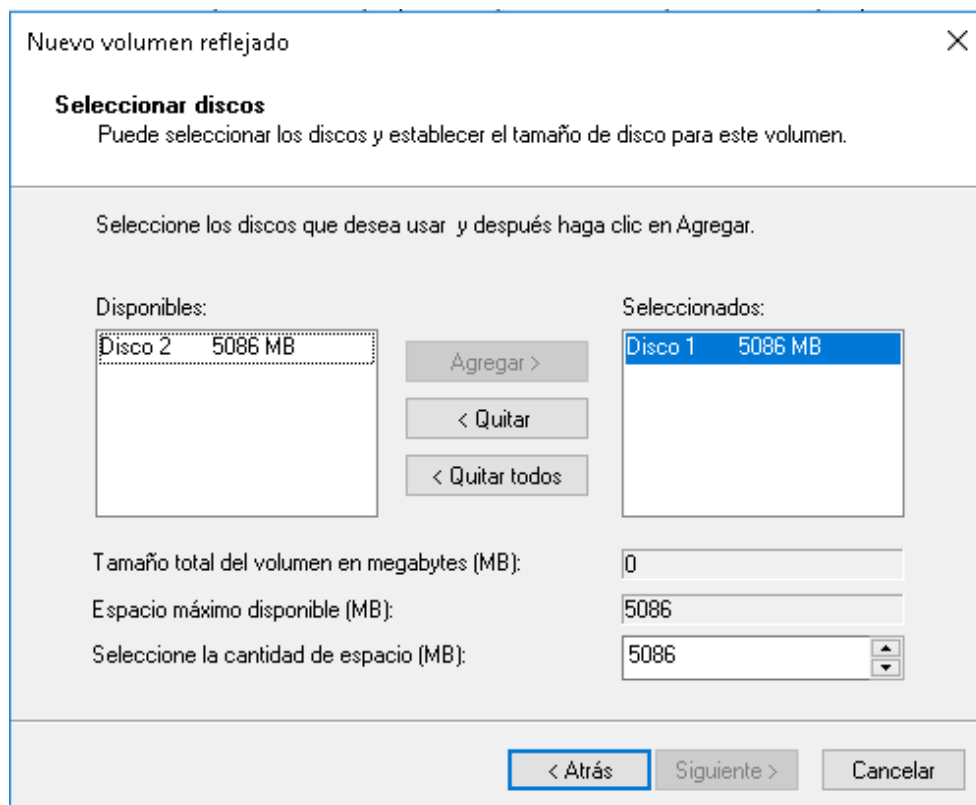


Imagen: generación de un volumen reflejado

Se muestra un asistente, en el que deberemos seleccionar los dos discos que deseamos convertir en RAID 1



Nuevo volumen reflejado [X]

Seleccionar discos
Puede seleccionar los discos y establecer el tamaño de disco para este volumen.

Seleccione los discos que desea usar y después haga clic en Agregar.

Disponibles:		Seleccionados:
Disco 2 5086 MB	Agregar >	Disco 1 5086 MB
	< Quitar	
	< Quitar todos	

Tamaño total del volumen en megabytes (MB): 0

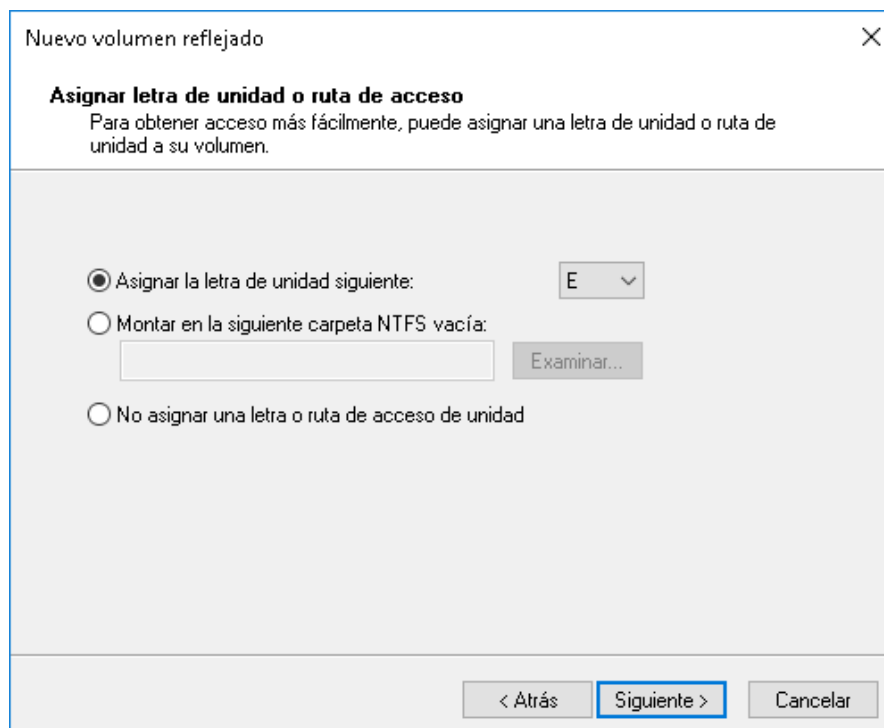
Espacio máximo disponible (MB): 5086

Seleccione la cantidad de espacio (MB): 5086

< Atrás Siguiendo > Cancelar

Imagen: selección de discos para el RAID 1

Se asigna una letra de unidad al nuevo volumen reflejado.



Nuevo volumen reflejado [X]

Asignar letra de unidad o ruta de acceso
Para obtener acceso más fácilmente, puede asignar una letra de unidad o ruta de unidad a su volumen.

☒ Asignar la letra de unidad siguiente: E

☐ Montar en la siguiente carpeta NTFS vacía: Examinar...

☐ No asignar una letra o ruta de acceso de unidad

< Atrás Siguiendo > Cancelar

Imagen: Asignación de letra de unidad

Se le da el formato deseado.

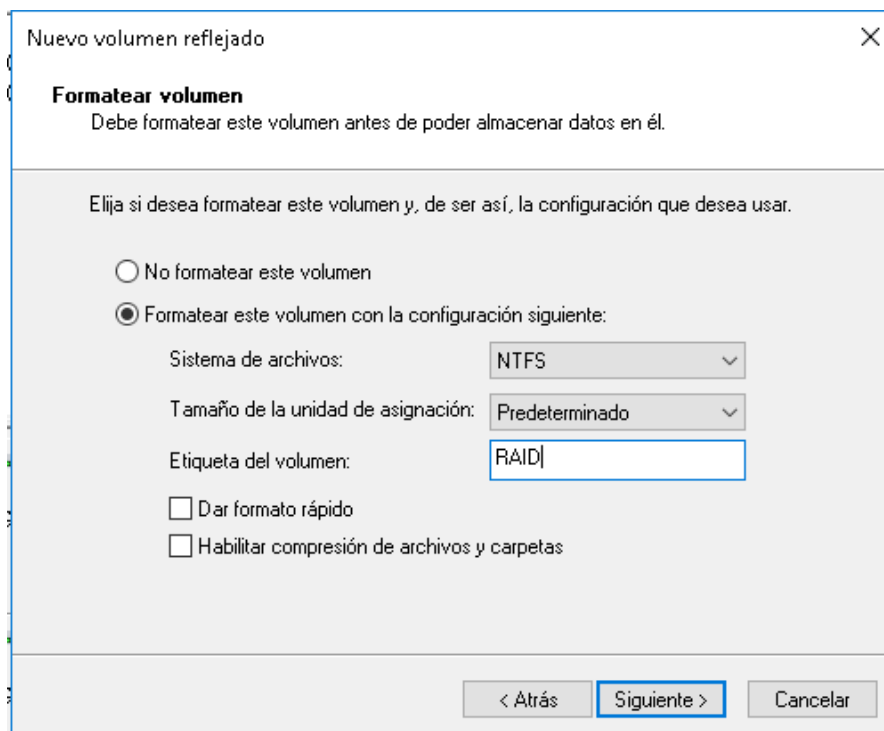


Imagen: formateo del nuevo volumen

Al finalizar el asistente, se nos avisa de que los discos deben convertirse en dinámicos. Se acepta.

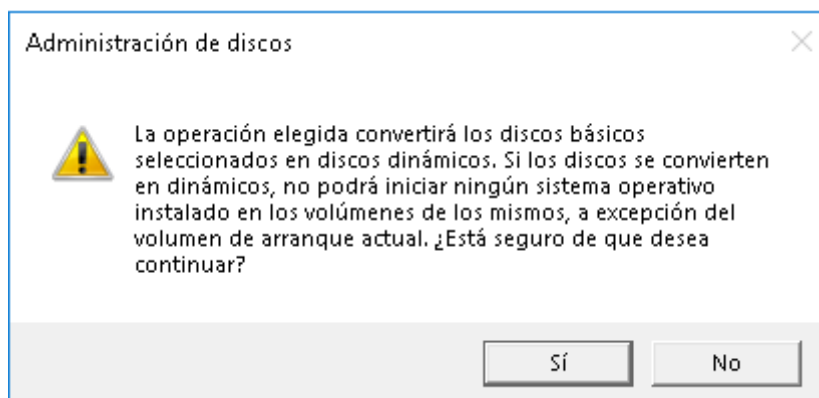


Imagen: conversión en discos dinámicos

Y ya se ha creado un sistema RAID 1

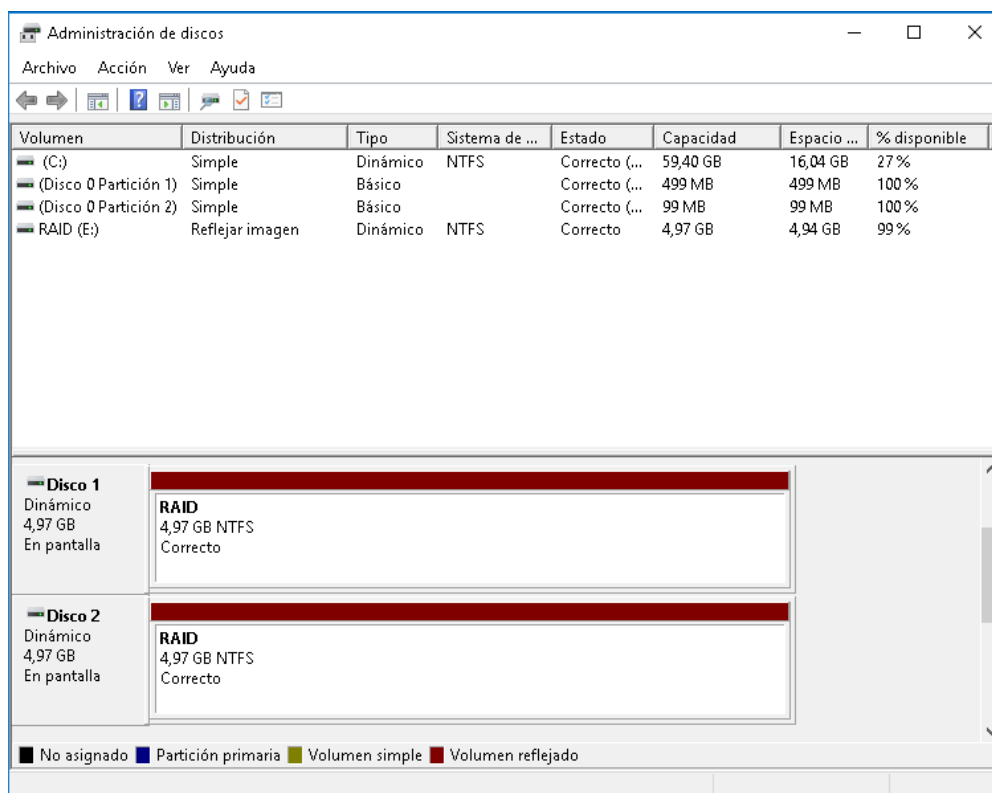


Imagen: RAID 1 creado

A partir de este momento, toda la información que se guarde en el disco RAID estará duplicada en los dos discos que se configuran en reflejo.

7. AUTOMATIZACIÓN DE TAREAS

En el trabajo diario con un sistema informático hay una serie de tareas que se repiten con frecuencia. Por ejemplo, la desfragmentación de discos o la realización de copias de seguridad, que se han tratado en apartados anteriores.

Todos los sistemas operativos proporcionan herramientas que permiten automatizar esas tareas frecuentes, de modo que se pueda planificar su realización **de forma automática**, para así poder prestar más atención al resto de actividades.

Por ejemplo, Windows 10 dispone del **Programador de tareas** para realizar dicha automatización. Podemos acceder a él a través del Panel de control -> Herramientas administrativas.

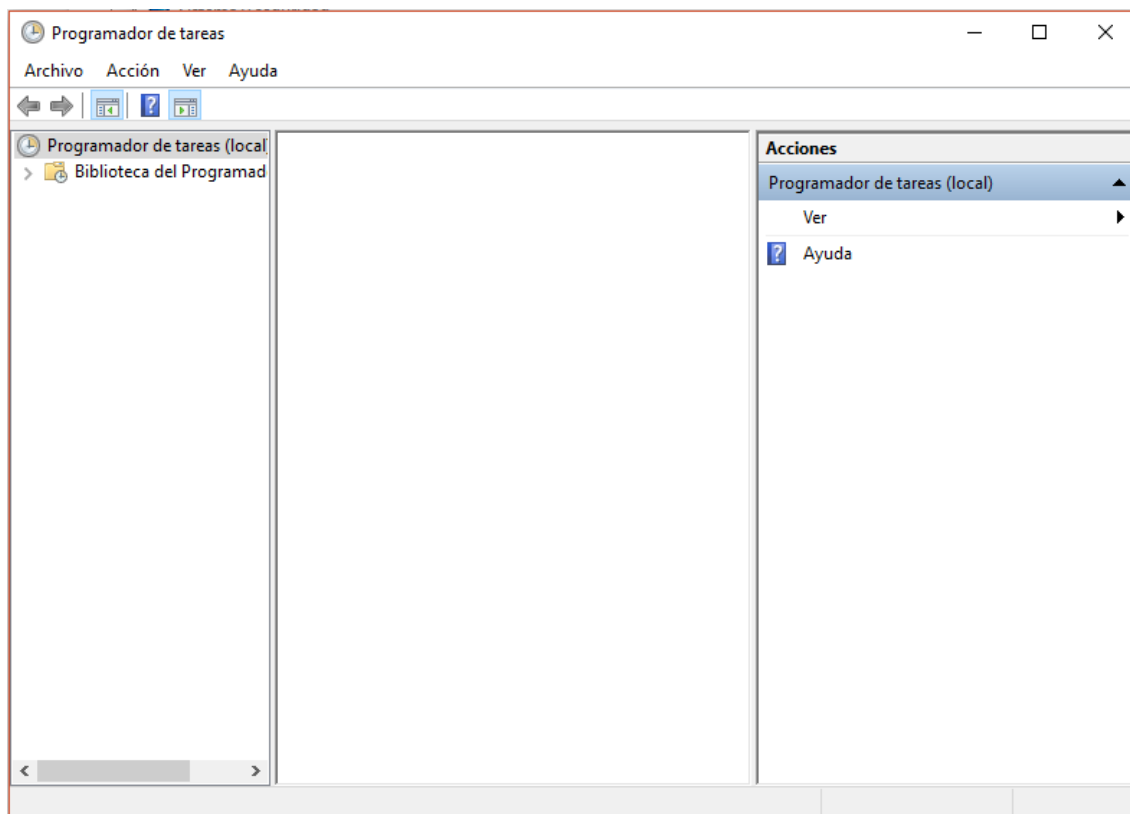


Imagen: programador de tareas en Windows 10

Para crear una nueva tarea, seleccionamos la opción “**Crear tarea básica**” del menú Acción. En este ejemplo se va a automatizar la desfragmentación de discos.

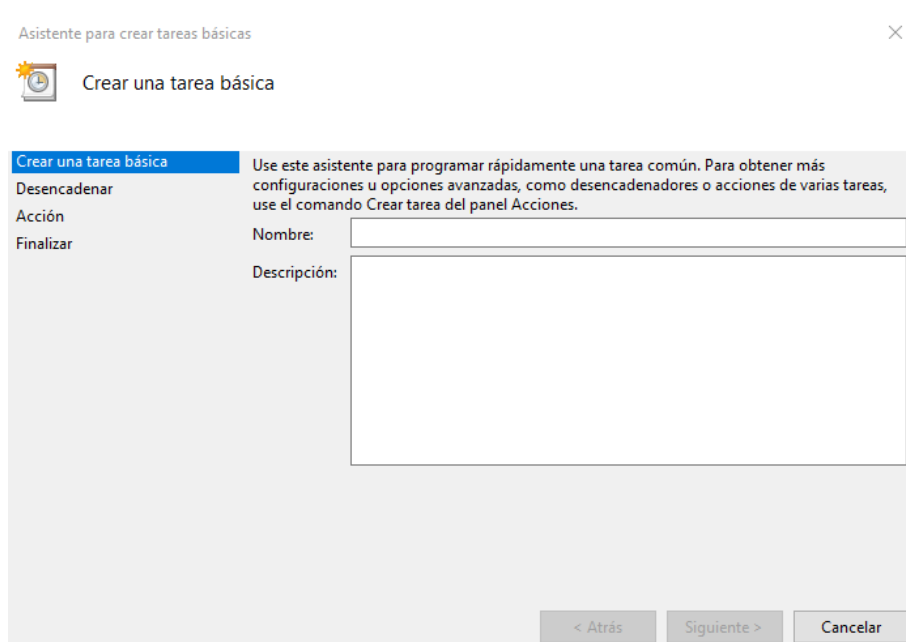


Imagen: asistente para crear una nueva tarea

En primer lugar habrá que proporcionar un nombre y una descripción para la nueva tarea.

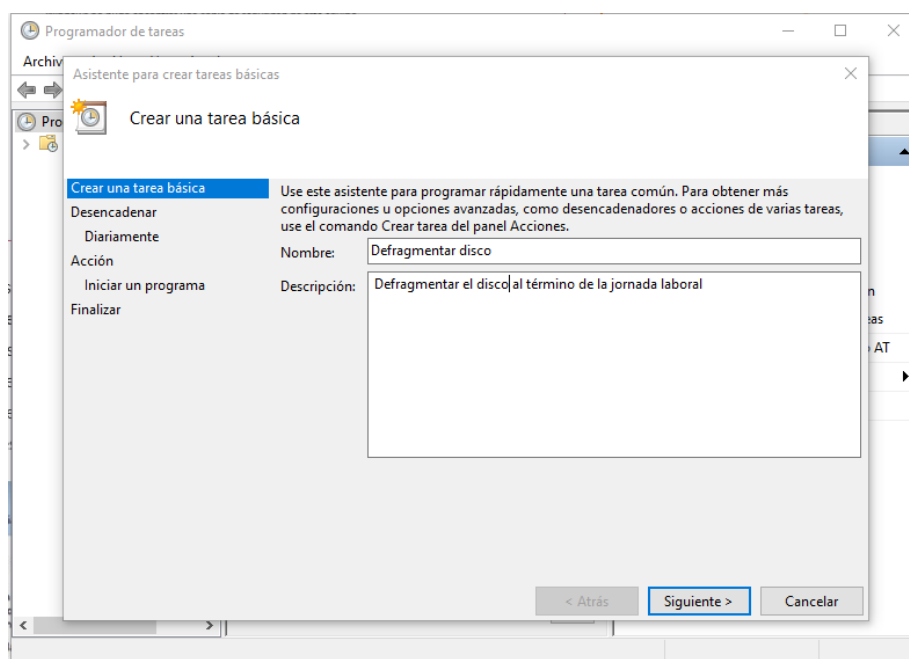


Imagen: asignación de nombre y descripción para la tarea

Se pulsa Siguiente y se accede a la sección **Desencadenar**. En la que habrá que indicar en qué momento se desea que se realice la nueva tarea. En este caso se desea que la desfragmentación se realice diariamente.

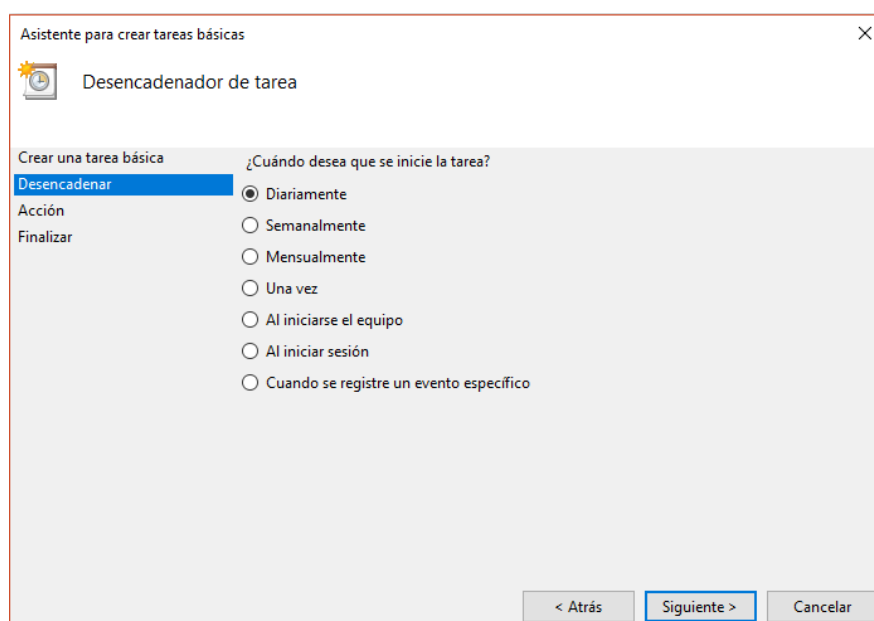
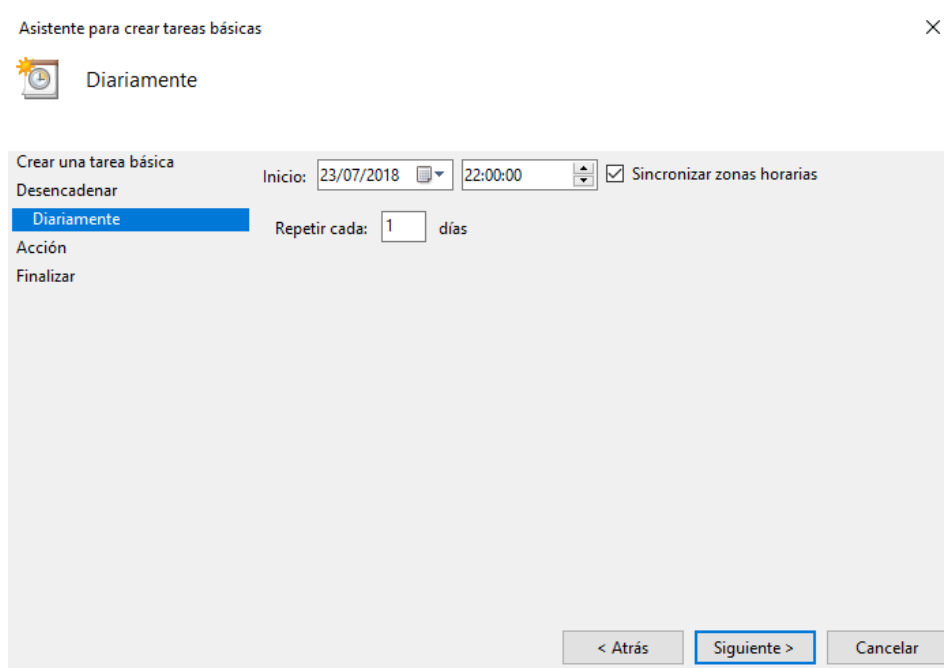


Imagen: establecimiento del desencadenador de la tarea

A continuación se establecen la fecha y hora de inicio para la realización de la tarea.



Asistente para crear tareas básicas

Diariamente

Crear una tarea básica

Desencadenar

Inicio: 23/07/2018 22:00:00 ☒ Sincronizar zonas horarias

Repetir cada: 1 días

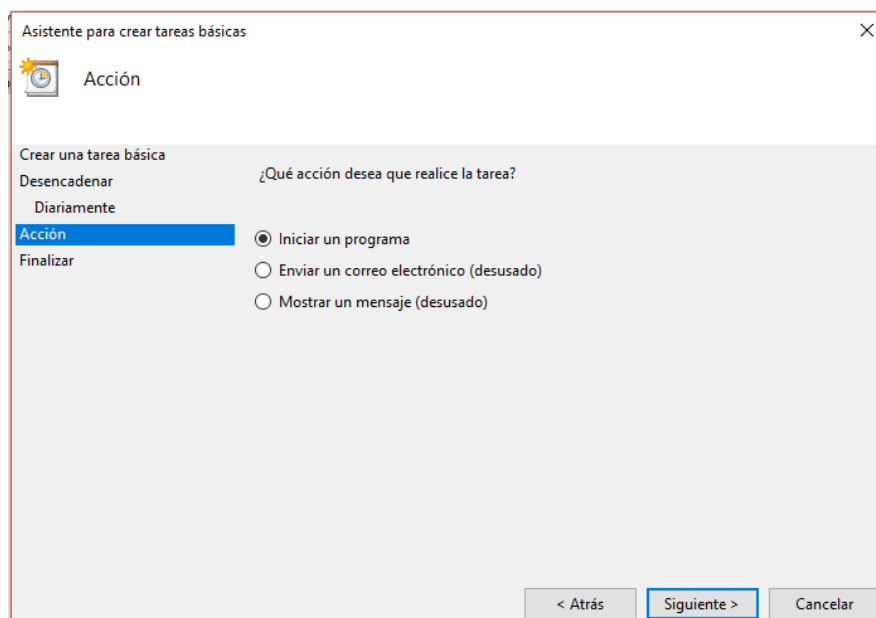
Acción

Finalizar

< Atrás Siguiete > Cancelar

Imagen: indicación de la fecha y hora de inicio

Al pulsar Siguiete se accede al tercer apartado del asistente, el que permite configurar la **Acción** que se desea llevar a cabo. En este caso se trata de Iniciar un programa, concretamente la herramienta de desfragmentación.



Asistente para crear tareas básicas

Acción

Crear una tarea básica

Desencadenar

Diariamente

Acción

Finalizar

¿Qué acción desea que realice la tarea?

☒ Iniciar un programa

☐ Enviar un correo electrónico (desusado)

☐ Mostrar un mensaje (desusado)

< Atrás Siguiete > Cancelar

Imagen: configuración de la acción a llevar a cabo

Se localiza el archivo ejecutable correspondiente al programa que se desea ejecutar de forma periódica. En este caso, Defrag.exe.

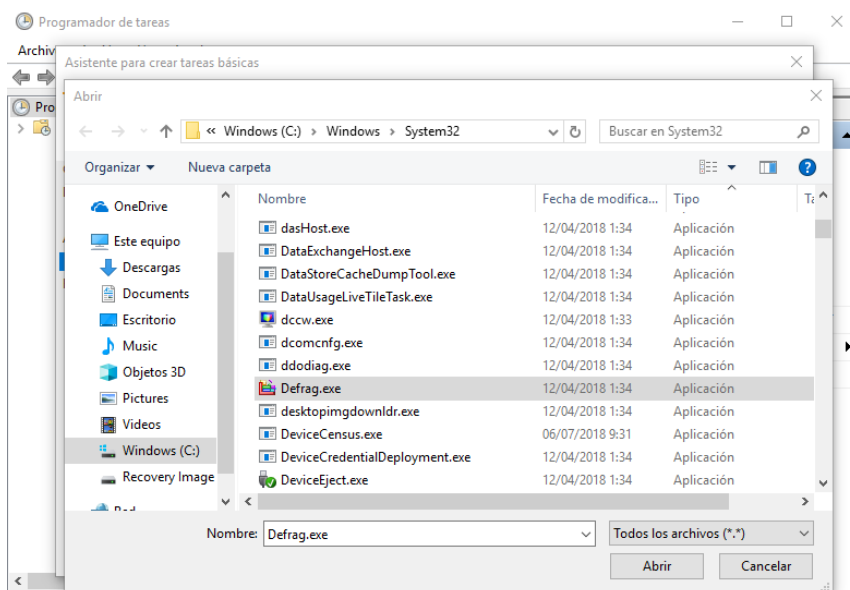


Imagen: selección del programa a ejecutar

Y con ello ya está configurada por completo la nueva tarea. El último apartado del asistente nos proporciona un **Resumen** de las opciones elegidas, y al pulsar el botón Finalizar se termina el proceso de creación.

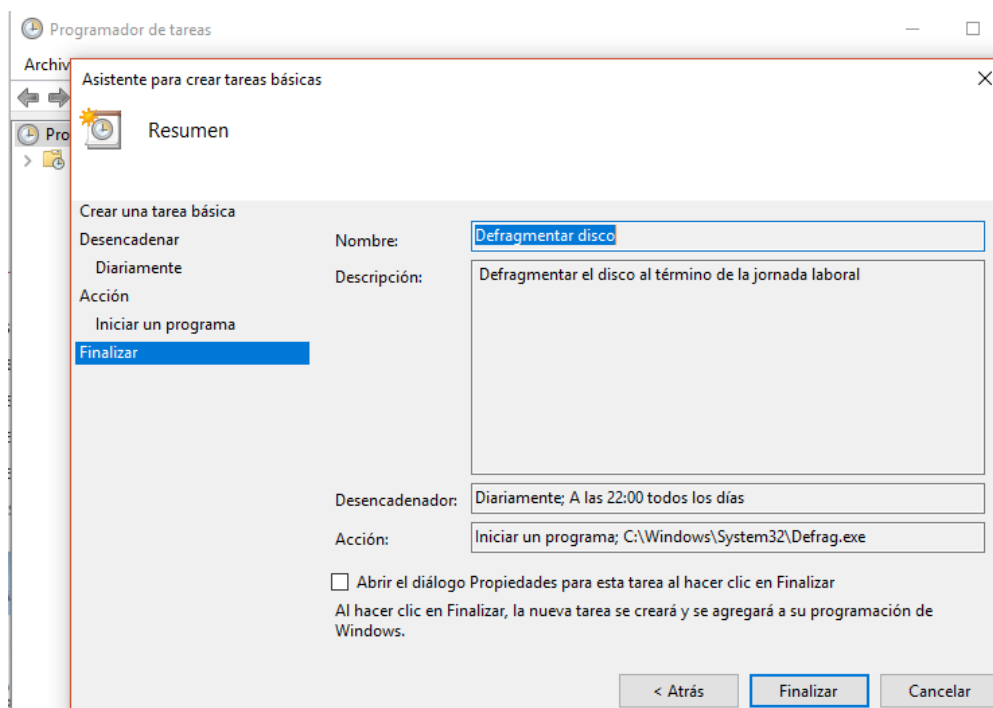


Imagen: finalización de la nueva tarea automatizada

RESUMEN FINAL

En esta unidad se ha comenzado por presentar el modo en que se estructura el almacenamiento de información en un sistema informático, introduciendo el concepto de sistema de archivos. A continuación se han descrito los sistemas de archivos de los principales sistemas operativos, y se han presentado los conceptos de archivo y directorio.

Seguidamente se han estudiado las principales tareas relacionadas con la gestión de la información, y se ha visto de forma práctica el modo de llevarlas a cabo en sistemas operativos libres y propietarios, tanto de forma gráfica como mediante comandos.

Se han identificado las variables de entorno y se ha trabajado la configuración de estas.

Por último, se han presentado técnicas básicas para la administración de discos, como la creación de particiones, la desfragmentación y las copias de seguridad. Así como técnicas avanzadas, como la implementación de RAID.

Y se ha mostrado el modo de automatizar las tareas repetitivas de nuestro sistema.