

## TEMA 5

### Sistema operativo propietario II

#### Guiones por lotes (20 horas)

##### Caso práctico



Nora se encuentra al arrancar el ordenador que no funciona correctamente, tiene que reparar el sector de arranque y necesita trabajar en modo texto. Menos mal que en la clase de Iván, han visto el símbolo del sistema de Windows, pues ha habido problemas con la instalación y cableado de red del aula. Para solucionarlo, todos los alumnos y alumnas han comprobado el funcionamiento de la red. Por un lado, han comprobado el cableado físico del aula y las conexiones al panel de parcheo y han tenido que usar comandos de red, para comprobar a qué grupo pertenece cada equipo y las direcciones IP que tenían asignadas. También han usado el comando **ping** para comprobar el funcionamiento de internet. Está acostumbrado a trabajar con comandos ya que cualquier administrador o administradora en Windows, se ha visto obligado a abrir la consola CMD, abreviatura de Command, comando en inglés, e introducir comandos para realizar alguna tarea que no era viable ejecutar por otros medios. Iván le enseña a Nora una serie de comandos básicos para poder solucionar su problema.

Esta interfaz nos recuerda al S.O. MS-DOS o **Sistema Operativo de Disco de Microsoft**, un Sistema operativo monousuario y monotarea, es decir, dedica los recursos del ordenador a un único usuario o usuaria y no puede procesar simultáneamente más de una aplicación. Una de las características principales es que permite hacer funcionar a un ordenador con pocos comandos u órdenes. Mientras que en otros sistemas operativos la shell, caparazón en inglés, o **consola** proporciona un medio potente y ágil para gestionar cualquier aspecto del funcionamiento del ordenador, en Windows no se le ha dado la suficiente importancia. Para mejorar esto, Microsoft ha creado la plataforma PowerShell.

- Windows 7 incorpora PowerShell, también viene instalado en Windows Server 2008 R2 y se puede instalar en versiones anteriores de Windows como Windows XP, Vista, Server 2003, Server 2008.
- Es compatible con sistemas de 32 bits y 64 bits.
- Los comandos u órdenes en PowerShell reciben el nombre de cmdlets.
- Es una consola orientada a objetos.
- PowerShell está basado en .NET Framework de Microsoft. .NET Framework es un conjunto de módulos de software, bibliotecas, librerías, programas, etc., que ayudan al desarrollo de otros programas.

- Permite acceso a almacenes de configuraciones como, por ejemplo, al registro de Windows. Se pueden ejecutar scripts *creados* en Windows Script Host. Windows Script Host ofrece la posibilidad de crear archivos por lotes complejos y potentes.
- Incorpora las clases del Instrumental de administración de Windows (Windows Management Instrumentation, WMI). WMI es un conjunto de funciones y procedimientos del sistema operativo Windows para controlar, monitorizar y administrar los equipos en una red.
- Ofrece un completo sistema de ayuda (hay un cmdlet que sirve como ayuda y es **Get-Help**).
- Permite utilizar la tecla tabulador para completar el nombre de un comando, fichero, carpeta, etc. En inglés se conoce como *Command Completion*.
- Se pueden crear y utilizar cmdlets nuevos, funciones y scripts.

Para acceder a PowerShell hay que pulsar Inicio - *Todos los programas* - *Accesorios* - *Windows PowerShell*.

Los cmdlets en PowerShell son sencillos de recordar porque usan el sistema *verbo- nombre* para llamar a los comandos: los verbos y los nombres están en inglés. Algunos ejemplos son: *Get-process*, *Stop-Process*, *Get-Date*, etc.

Los cmdlets usan parámetros que son propios, esto quiere decir que los parámetros de un cmdlet pueden no ser los mismos que los de otro, aunque en general son los mismos. Algunos ejemplos de parámetros son: *-Name*, *-Path*, *-ComputerName*, etc.

PowerShell tiene la posibilidad de crear alias, un alias permite renombrar o llamar a los comandos de distintas formas. Esto es muy útil porque se pueden crear alias con nombres de parámetros que utilicemos en otros sistemas operativos o que nos inventemos nosotros mismos, por ejemplo, podemos utilizar el alias *ps* para listar los procesos, este comando es propio de Linux: también podemos crear un alias inventado por nosotros que sea *Dime-Procesos*.

## Comandos básicos del símbolo del sistema.

### Caso práctico



Iván cuenta a Nora que tiene que hacer las prácticas en una gran fábrica de su ciudad, y que el servidor de la fábrica sólo trabaja en modo texto ya que al dar servicio a todos los equipos de la fábrica no debe consumir muchos recursos, es decir, sólo trabaja mediante comandos, aunque en el resto de equipos los usuarios y usuarias trabajan en modo gráfico. Está muy contento pues va a poder practicar con lo que ha visto en el aula.

A las interfaces en modo texto también se les llama CLI(Command Line Interface) o interfaz en modo comando. A las interfaces en modo gráfico se les llama GUI (Graphic User Interface) o interfaz en modo gráfico. Para entrar en modo texto, elegimos en el menú Inicio, la opción Ejecutar y escribimos cmd o en el menú Inicio, en Accesorios, elegimos Símbolo del Sistema.

En esta unidad te vas a olvidar del ratón y vas a trabajar en modo comando. **Un comando o instrucción es cada una de las órdenes que un operador u operadora da al ordenador para que ejecute una acción....(command).**

Dicho de otro modo, un comando es una orden que se le da a un programa de computadora que actúa como intérprete del mismo, para así realizar una tarea específica.

Generalmente un comando se le da a una interfaz de línea de comandos, como un shell. En Windows puedes realizar casi cualquier tarea desde el entorno gráfico, pero siempre habrá momentos en que nos será necesario saber movernos por la interfaz en modo texto. Hay algunos comandos que no funcionan en Windows XP y otros que no funcionan en Windows 7, aunque la mayoría funcionan por igual en ambos. Para ver algunos de los comandos disponibles puedes escribir el comando HELP. Verás que muchos comandos tienen modificadores o permiten argumentos, es decir, información que se le pasa a un comando para que cambie su comportamiento.

La sintaxis general de un comando es: **COMANDO [unidad lógica][/parámetro1][/parámetro2]**



Lo que se indica entre corchetes es opcional y puedes obviar utilizarlo.

Te voy a mostrar comandos concernientes al S.O.

**HELP** [comando]: Muestra la descripción, sintaxis y modificadores de cualquier orden.

**PROMPT** [texto][\$modificadores]: Comando que nos indica la unidad y directorio activo.

**CLS**: Este comando limpia la pantalla, tras ejecutarlo sólo quedará en la pantalla el prompt.

**DATE** [fecha(según formato)]: nos muestra la fecha del sistema y nos permite cambiarla. [horas: minutos segundos[centésimas]]: Nos muestra la hora del sistema y nos permite modificarla.

**VER**: Nos permite ver la versión del sistema operativo que estamos utilizando.

Cuando queramos información de un comando también usaremos el comando HELP seguido del comando que tengamos duda o bien escribimos el nombre del comando seguido de /?

*Un comando es una orden o una instrucción, que sigue una sintaxis, y para ejecutarle necesitamos hacerlo desde la consola o shell...*

En este enlace puedes ver una página muy completa con los comandos con los que trabajaremos.

[Comandos](#)

## Caracteres comodín.

\*

?

Ya sabrás que los nombres de los ficheros pueden constar de una o dos partes, la primera hace referencia al nombre del fichero y la segunda es la extensión, estas dos partes van separadas por un punto, es decir, **nombre.extensión**.

Los archivos tienen siempre:

- Nombre: puede tener hasta 255 caracteres
- Extensión: tres caracteres, hace referencia al tipo de documento, \*.exe, \*.txt, \*.doc, \*.exe, etc

Te habrás preguntado que hacen los caracteres comodín, lo que hacen es sustituir a un carácter, o a varios, en los nombres de fichero, lo que te va a permitir agrupar órdenes. Los caracteres comodines en MS-DOS son los siguientes:

1. **Carácter ?**. Representa a un carácter válido en el nombre de un fichero. Sólo sustituye o representa unos de ellos.
2. **Carácter \***. Representa uno o más caracteres válidos en el nombre de un fichero. Sustituye caracteres por el principio, por el final o por el centro del nombre de archivo especificado.

A continuación te muestro unas plantillas típicas para el manejo de nombres de ficheros:

- **\*.EXE**. → Todos los ficheros con extensión EXE.
- **\*\*\*\*\*.BAT** → Todos los ficheros con un nombre de cinco caracteres y extensión BAT.
- **\*.??** → Todos los ficheros con cualquier nombre y longitud, y con sólo dos caracteres de extensión.
- **\*.\*** → Todos los ficheros.
- **\*\*\*\*\*.???** → Todos los ficheros.

La mayoría de los comandos de administración de ficheros que pueden incluir trayectorias aceptan caracteres comodín, a excepción del comando TYPE.

Cuando abres la terminal, la información que te aparece justo antes de poder teclear comandos es el prompt, se muestra por defecto situado en el directorio del usuario. `C:\> C:\primera\>`

Visita el siguiente enlace para saber más sobre MS DOS. Es un curso sobre conceptos básicos y avanzados de MS-DOS.

[MS DOS.](#)

## Comandos básicos sobre ficheros.

Vas a ver una lista de comandos que te serán útiles cuando quieras trabajar con ficheros. Si quieres ver su sintaxis y todas las opciones posibles de cada comando, puedes recurrir al comando **HELP**.



- **TYPE**: Muestra el contenido de un archivo por pantalla. Generalmente sólo aparecerán legibles los archivos de texto.
- **PRINT**: El funcionamiento es similar al de TYPE, con la diferencia de que se pueden imprimir varios ficheros a la vez.

- **RENAME (REN):** Cambia el nombre a uno o más ficheros. Este comando admite el uso de comodines.  
No puede existir previamente en el directorio otro fichero con el mismo nombre que se asignará al fichero que se va a renombrar. Si existe trayectoria para el fichero original, ésta será la del nuevo fichero. La trayectoria de destino del nuevo fichero no existe. Es decir, el nuevo fichero se crea en la trayectoria especificada del primero.
- **COPY:** Copia uno o más archivos a otro lugar. Si no se especifica la trayectoria de destino, la copia se realizará en el directorio y unidad predeterminados (activos) y tendrá el mismo nombre que el archivo original. Si no se indica la ruta, se copiarán los ficheros que se encuentren en la unidad y directorio actuales.
- **MOVE:** El comando MOVE mueve uno o varios archivos de un directorio a otro. Este comando también permite el uso de comodines cuando queremos mover varios archivos a la vez.
- La diferencia entre MOVE y COPY es clara; MOVE sólo mantiene una copia del archivo en la estructura. COPY permite mantener dos copias, el original y el archivo nuevo.
- **DEL** Con este comando y con ERASE, podemos eliminar archivos dentro de la estructura jerárquica. Pero no se puede utilizar para eliminar directorios.
- **ATTRIB:** Muestra o cambia los atributos de un archivo. Un archivo puede tener los siguientes atributos:
  - R: Atributo de sólo lectura (Read, en inglés), protege a un fichero de forma que sólo puede ser leído.
  - H: Oculto (Hidden, en inglés), para ocultar un fichero.
  - A: Atributo de Archivo, sirve para saber si un fichero ha sido modificado.
  - S: de Sistema, para identificar los archivos propios del Sistema.
- **XCOPY:** Este comando tiene un funcionamiento parecido al de COPY, con la diferencia de que además de copiar uno o varios archivos, permite copiar estructuras completas de directorios. Con este comando no se puede copiar una estructura de ficheros y directorios en otro lugar en el que esté contenido algún directorio que sea el origen de la estructura que vamos a copiar.

```
Cls
Copy con texto1.txt (Ctrl+z)
Copy con texto2.txt
Copy con texto3.txt
Copy con prueba.txt
Md c:\textos\primero
Copy c:\*.txt c:\textos\primero
Dir *.txt
Dir c:\textos\primero\t*.txt
Dir c:\textos\primero
Cd textos\primero
del p.*
dir
cd\
del c:\textos\primero\*.txt
copy texto1.txt texto2.dup
dir
type c:texto1.txt
tu
type *.txt
copy c:\textos\uno.bas \apuntes\
move c:\libros\autor.dat datos.dat
attrib autoexec.bat +h+r
attrib -h prueba.txt
xcopy prueba:txt a:
xcopy prueba:txt a:\textos\
xcopy prueba:txt a:\textos\asunto.txt
xcopy prueba:txt
move c:\almacen c:\datos
move c:\textos\texto1.txt c:
```

## Comandos de información del sistema.



Hay muchos más comandos, ahora vas a ver una serie de comandos que se usan a menudo en el testeado del equipo:

- **DXDIAG**: Lanza la herramienta de diagnóstico de Direct X, permitiendo comprobar lo referente a los controladores.
- **MSCONFIG**: Lanza la utilidad de configuración del sistema, con lo que podremos seleccionar programas que se carguen durante el inicio de Windows.
- **REGEDIT**: Nos lanza el editor del registro.
- **SYSTEMINFO**: Muestra información sobre nuestro equipo.
- **IPCONFIG**: Nos muestra y permite renovar la configuración de todos los interfaces de red.
- **PING**: Comprueba si otra máquina está encendida.
- **ROUTE**: permite ver las tablas de enrutamiento.
- **RUNAS**: Permite ejecutar un programa con privilegios de otra cuenta.

## Organización de la información.

Caso práctico



Nora ya ha reparado su equipo con ayuda de Iván. Va a aprovechar y cambiar cosas de la configuración del equipo y testear unidades de almacenamiento usando la consola. Ha detectado que tiene un virus en su pendrive y con el antivirus no es capaz de eliminarle ni si quiera desde el entorno gráfico. Por tal motivo, ha accedido desde la terminal al pendrive moviéndose por los directorios. A continuación ha hecho un listado de todos los ficheros ocultos y ha eliminado al virus usando comandos.

El sistema operativo se caracteriza por tener una estructura jerárquica. Toda la información que gestiona, sea del propio sistema operativo o información de usuario o usuaria, se almacena en forma arborescente. Para ello, se utilizan los directorios, los cuales, a su vez, pueden contener subdirectorios y ficheros o archivos. En este apartado vas a usar los comandos de los puntos anteriores y verás la organización de la información en tu sistema. Aprenderás a crear estructuras jerárquicas y a manejarte con ellas.

## Unidades, ficheros y directorios.

Esta estructura arborescente parte de la propia unidad en la que vayas a almacenar los datos. Las unidades son los dispositivos de almacenamiento. Así, se habla de una unidad de disco duro, unidad de disquete, unidad de CD-ROM, primera partición, segunda partición, etc. Estas son **unidades físicas**. Para realizar la comunicación entre las unidades físicas y el usuario o usuaria se utiliza una relación. Esta relación es la que permite al usuario o usuaria reconocer las unidades físicas gracias a las **unidades lógicas**. Las unidades lógicas son la representación que el sistema operativo hace de cada unidad física. Las unidades lógicas se representan mediante una letra del alfabeto, seguida de dos puntos (:). La relación entre unidades lógicas y físicas se muestra en la siguiente tabla:

Asociación entre unidades lógicas y físicas

Unidad lógica	Unidad física
A:	Primera unidad de disquete.
B:	Segunda unidad de disquete.
C:	Primera unidad de disco duro. Segunda unidad de disco duro.
D:	Primera unidad de CD-ROM en ausencia de la segunda unidad de disco duro. Segunda partición del primer disco duro.
E:	Siguiente unidad de disco duro, de CD-ROM, tercera partición de disco duro, etcétera.
F:	Siguiente unidad de almacenamiento

Seguramente ya sabes que las unidades A: y B: hacen referencia a unidades de disco flexible (disquete). Las siguientes unidades lógicas se asignan a partir de la C: Desde esta unidad lógica en adelante, quedarán referenciados todos los dispositivos de almacenamiento como discos duros o CD-ROM o particiones de disco duro siempre y cuando tengamos más de una. La forma de almacenar información en las diferentes unidades dependerá del tipo que ésta sea. Pueden ser archivos regulares o estándares, directorios o subdirectorios. Te voy a explicar cada uno de estos componentes:

- **Un directorio es una zona de cualquier unidad de almacenamiento destinada a almacenar ficheros o archivos regulares y otros directorios o subdirectorios.** En esta zona se organiza el almacenamiento de datos.
- **Un subdirectorio es un directorio dentro de otro directorio** que se utiliza para almacenar ficheros o archivos, u otros subdirectorios. Es similar al directorio, pero jerárquicamente está por debajo de él.
- **El fichero o archivo es el propio elemento de almacenamiento de la información.** Los ficheros pueden ser de varios tipos, y son la esencia del almacenamiento de la información en los soportes.

*La estructura es arborescente y jerárquica.* Todo cuelga del directorio raíz. Cada directorio, una vez creado, está vacío, es decir, no tiene ni subdirectorios ni archivos o ficheros que cuelguen de él. Pero al crear un directorio, se genera automáticamente, dentro de él, dos subdirectorios:

- . (punto). Este subdirectorio hace referencia al contenido del propio directorio.



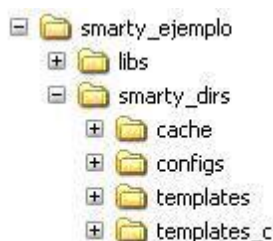
- .. (dos puntos). Este subdirectorio hace referencia al directorio padre, que es aquel que se encuentra jerárquicamente por encima del directorio en el que estamos situados actualmente. Por tanto, el directorio raíz no contendrá este fichero, ya que por encima de él no hay ningún otro directorio en la jerarquía.

## 1. Trayectorias y caminos.

Existen distintas formas de hacer referencia a un fichero o directorio, dependiendo de la parte de la estructura jerárquica donde estamos situados. Para empezar a explicar cómo se accede a los ficheros y directorios, hemos de tener en cuenta dos conceptos:

- Unidad activa.
- Directorio activo.

Se llama **unidad activa** a la letra que indica la unidad lógica con la que se corresponde una unidad física sobre la que el sistema operativo tiene el control o está situado en ese momento. Para poder acceder a un fichero o directorio concreto hay que saber en qué unidad estamos situados, es decir, cuál es la unidad activa o **directorio activo**.



Es el propio sistema operativo el que indica cuál es el directorio activo, gracias al denominado símbolo del sistema. Lógicamente, según el usuario o usuaria cambie de un directorio a otro, o de una unidad a otra, el directorio activo y la unidad activa cambiarán. En cualquier caso, estemos donde estemos, siempre existirá unidad activa y directorio activo.

Partiendo de una unidad activa y de un directorio activo, la posición de un fichero o un directorio en nuestro disco se denomina **trayectoria**. *Una trayectoria o ruta es un nombre que indica la posición de directorios y ficheros.* Por otro lado, las trayectorias no afectan a las unidades lógicas. La unidad lógica será A: B: C:

Una trayectoria o ruta es la que se construye partiendo desde donde estamos y bajando por la estructura jerárquica para llegar al directorio o fichero que deseamos. Dentro de las trayectorias, es necesario tener en cuenta el carácter \. Este símbolo sirve para separar los nombres de directorios y subdirectorios.

Hay dos tipos de trayectorias:

- **La trayectoria absoluta:** nos indica la dirección de un fichero desde el directorio raíz, sea quien sea el directorio activo. Por tanto, una trayectoria absoluta se construye indicando en primer lugar la unidad, seguida de \, para seguir con una cadena de subdirectorios, cada uno de ellos separados por \, hasta llegar a indicar exactamente dónde se encuentra el archivo.

- **La trayectoria relativa:** para utilizar estas trayectorias es importante tener en cuenta tanto el directorio activo como la unidad activa. Es lógico pensar, al contrario que en la trayectoria absoluta, que será única, que las trayectorias relativas serán múltiples. La trayectoria sea diferente dependiendo del directorio y la unidad activa en la que nos encontremos. La trayectoria relativa nos indica la dirección de un fichero desde el directorio activo.

## Comandos básicos sobre directorios.



¿Te imaginas un armario sin baldas ni cajones? No habría forma de encontrar nada. En un disco duro ocurre lo mismo, necesitamos poder organizar la información en diferentes sitios. Con los directorios puedes almacenar información de forma ordenada. Es una forma de estructurar la información almacenada en nuestro soporte de almacenamiento. Los nombres de directorio siguen las mismas reglas de formación que los nombres de fichero, pero no suelen tener extensión, aunque puede tenerla. Con los ficheros ocurre lo mismo: pueden tener un nombre con extensión o sin ella, aunque los archivos suelen tener extensión. Vas a ver una lista de comandos que te serán útiles cuando quieras ver el contenido de un fichero, copiar ficheros, eliminarlos, moverlos y moverlos por la estructura jerárquica. Cuando quieras ver la sintaxis y todas las opciones posibles de cada comando recurriremos al comando **HELP**.

- **DIR** (Muestra el contenido de un directorio, informando sobre el nombre y extensión de cada archivo, el tamaño en bytes, la fecha y hora en que se creó o modificó por última vez, y los subdirectorios que cuelgan de él. Indica también al principio de la lista, y como información complementaria, la etiqueta y número de serie y de disco, y al final indica el número y tamaño del conjunto de archivos que se han mostrado y el espacio que queda libre en el disco.
- **CHDIR o CD (de Change Directory)**: Con este comando podemos movernos por la estructura de directorios creada. Sirve para subir o bajar por la estructura jerárquica. Si ejecutamos el comando CD sin parámetros, muestra el directorio y la unidad activa en la que estamos situados.
- **MKDIR o MD (Make directory)**: Crea directorios por debajo del actual.
- **RMDIR o RD (Remove Directory)**: Con este comando podemos eliminar directorios en cualquier parte de la estructura jerárquica. Para borrar directorios hemos de tener en cuenta que sólo se puede borrar un directorio si está vacío.
- **MOVE**: Con este comando puedes cambiar el nombre de un directorio por otro. Es evidente que el directorio tiene que existir, y da igual dónde esté situado.
- **TREE**: muestra la estructura de directorios y subdirectorios contenida en cada uno de ellos de forma arborescente. Si se especifica un directorio, el comando mostrará la estructura que cuelgue de éste, no los que puedan existir por encima de él.

**DIR** [unidad:][ruta][archivo] [/A[:][atributos]] [/B] [/C] [/D] [/L] [/N] [/O[:][orden]] [/P] [/Q] [/R] [/S] [/T[:][fecha]] [/W] [/X] [/4]

[unidad:][ruta][nombre de archivo]

Especifica la unidad, el directorio y los archivos que se mostrarán.

/A Muestra los archivos con los atributos especificados.

atributos D Directorios R Archivos de solo lectura

H Archivos ocultos A Archivos para archivar

S Archivos de sistema I No archivos indizados de contenido

L Puntos de análisis - Prefijo de exclusión

/B Usa el formato simple (sin encabezados ni sumarios).

/C Muestra el separador de miles en el tamaño de los archivos. Es la opción predeterminada. Use /-C para deshabilitar la aparición de dicho separador.

/D Similar al listado ancho, pero los archivos aparecen clasificados por columnas.

/L Usa letras minúsculas.

/N Nuevo formato de lista larga donde los nombres de archivo aparecen en el lado derecho.

/O Muestra los archivos según el orden indicado.

Orden N Por nombre (orden alfabético)

E Por extensión (orden alfabético)

S Por tamaño (orden creciente)

D Por fecha y hora (el más antiguo primero)

G Agrupar primero los directorios

- Prefijo para invertir el orden

/P Hace una pausa después de cada pantalla completa de información.

/Q Muestra el propietario del archivo.

/R Muestra las secuencias alternativas de datos del archivo.

/S Muestra los archivos del directorio especificado y todos sus subdirectorios.

/T Controla el campo de fecha que se mostrará o usará para realizar la clasificación

fecha C Creación

A último acceso

W última modificación

/W Use el formato de listado ancho.

/X Muestra los nombres cortos generados para los nombres de archivo sin formato 8.3. El formato es el mismo que para /N, con el nombre corto especificado antes del nombre largo. Si no existe un nombre corto, se muestran espacios en blanco en su lugar.

/4 Muestra el año con 4 dígitos

Los modificadores se pueden preestablecer en la variable de entorno DIRCMD. Estos se pueden invalidar precediendo cada modificador con - (guión), por ejemplo: /-W.

**MD directorio**

**Md c:\pruebas\almacen**

**Rd textos**

**cd..**

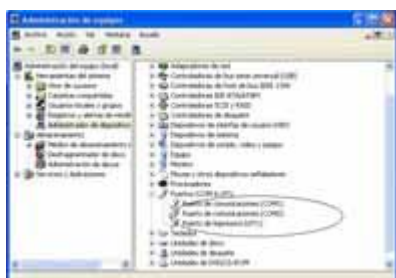
**cd ..\..**

**cd ..\prueba**

**cd \**

**tree**

## Conceptos de redireccionamiento.



Tienes que saber que los ficheros pueden ser de entrada o de salida. Si necesitas crear un fichero, es decir, introducir datos en un fichero, tendrás que utilizar los periféricos de entrada necesarios. Si lo que quieres es ver o imprimir un fichero, tendrás que utilizar los correspondientes periféricos de salida. De forma predeterminada, el periférico de entrada es el teclado, y el de salida, el monitor.

Genéricamente, los dispositivos que reconocen los sistemas operativos son los siguientes:

- **CON.** (abreviatura de Consola) Salida por pantalla.
- **PRN.** Salida por la impresora por defecto.
- **LPT1, LPT2 y LPT3.** Puertos paralelos en los que normalmente hay conectadas impresoras.
- **AUX, COM1, COM2, COM3 y COM4.** Puertos serie en los que hay conectado cualquier tipo de dispositivo periférico.
- **NUL.** Dispositivo nulo. Se utiliza cuando el usuario o usuaria quiere enviar información a un sitio sin que quede reflejo de ella. También sirve para simular una entrada ficticia.
- 

Cada dispositivo lleva asociado un nombre que no es modificable y que no puede ser adoptado por ningún nombre de archivo ni directorio. En algunas ocasiones puede ser interesante o incluso necesario cambiar la entrada o salida predeterminada. Para ello, lo que harás será redireccionar a otro dispositivo de entrada o salida. Las redirecciones utilizadas son las que te muestro en el siguiente esquema:

Redireccionamiento		
Símbolo	Descripción	Sintaxis
<	Redirección de entrada	Orden
>	Redirección de salida	Orden>fichero/dispositivo

### Redirección de entrada <.

Te habrás preguntado alguna vez si puedes cambiar la salida de algún comando, pues esto lo vas a poder hacer mediante la redirección de entrada. Consiste en cambiar a otro periférico o fichero el dispositivo de entrada estándar. Si ejecutamos el comando DATE, el sistema operativo esperará que el usuario o usuaria escriba por la entrada estándar la nueva fecha.



Otros comandos especial mente diseñados para esta función. Son los siguientes:

- **SORT.** Permite ordenar los datos de entrada.
- **MORE.** Muestra el contenido de los datos de entrada, dividiendo su presentación en pantallas completas.
- **FIND.** Busca una cadena de caracteres determinados dentro de los datos de entrada.

### Aplicaciones:

- Mostrar un fichero de texto con pausa en cada pantalla.

F:\>more

(More toma como entrada el fichero RECUERDO.TXT y como salida, el monitor)

- Ordenar un fichero alfabéticamente y verlo en la pantalla.

F:\>sort

(Sort toma como entrada el fichero PERSONAL.TXT y como salida, el monitor)

- Ordenar un fichero alfabéticamente a partir de una columna. Ejemplo: Si el fichero PERSONAL.TXT contiene nombres y datos del personal de una empresa, podemos ordenarlo por la columna que indica el puesto ejercido por cada empleado.

F:\>sort /+40

- Encontrar las filas que contienen una cadena de caracteres.

F:\>find "JIMENEZ"

(Find considera como entrada el fichero PERSONAL.TXT y muestra por pantalla todas las filas que contienen la cadena «JIMENEZ»)

### Redirección de salida >.

Vas a practicar el redireccionamiento de salida, le utilizarás especialmente para dos operaciones:

- Enviar la salida de un programa a la impresora.
- Enviar la salida de un programa a un fichero.

En el primer caso si quieres redireccionar la salida de una orden concreta a un dispositivo diferente a la salida estándar (por ejemplo, la impresora), bastará con escribir el comando determinado y, a continuación, escribir >LPT1 o PRN. De esta forma, se consigue que la información se imprima y no se muestre. Existe un medio de redireccionamiento de información, es decir, conseguir que la información que se desea ver, sea enviada a otro dispositivo. Para esta tarea se emplea el símbolo '>' seguido del nombre del dispositivo deseado.

## Ejemplo:

C:\> dir > prn

Obtendríamos una copia por la impresora del contenido del directorio actual, siempre y cuando la impresora estuviera conectada y en condiciones ideales para impresión.

En el segundo caso si deseas que el resultado de la ejecución de un comando concreto se almacene en un fichero, tienes que escribir el comando y, a continuación, escribir >nombre fichero.

Te crea un fichero de nombre hola1.txt y contenido hola, si el fichero hola1.txt tenía algún contenido con esto se sobrescribe con lo que pierdes el contenido anterior.

## Adición a un fichero >>.

Este tipo de redireccionamiento (append) le usarás exclusivamente para añadir más información a un mismo fichero. Cada vez que se redirecciona con la adición el mismo fichero, se aumenta su contenido. Tienes que tener en cuenta que tiene lo que pongas antes del redireccionamiento ha de comenzar por un comando.

Te voy a mostrar dos ejemplos:

echo adiós >>hola1.txt

La línea anterior añade al fichero hola1.txt una nueva línea con la palabra adiós.

En caso de no existir el fichero hola1.txt, le crea.

dir >>hola1.txt

Añade al fichero anterior hola1.txt, el resultado del comando dir.

Te permite añadir contenido a los ficheros desde la terminal, sin necesidad de editar el fichero.

## Ficheros de comandos o guiones por lotes.

### Caso práctico

Iván está realizando las prácticas en una fábrica de su ciudad. El tutor que le han asignado para realizar el seguimiento de sus prácticas le pregunta si será capaz de automatizar una serie de tareas en el servidor de la fábrica, que sólo trabaja en modo texto, ya que tiene que generar para cada trabajador una zona de trabajo y un fichero personal. ¡No tiene más remedio que ponerse a pensar en hacer archivos por lotes que le faciliten la tarea!

Ya sabrás que en los entornos de línea de comandos, es habitual ver las siguientes expresiones: un archivo "**por lotes**"; un archivo "**batch**"; un "script" o un archivo "**bat**". Se trata de diferentes formas de referirnos a lo que denominaremos archivos de comandos, ficheros por lotes, ficheros

batch. Te voy a explicar lo que son estos archivos. Son archivos de texto que permiten ejecutar diversos comandos del Shell y vas a aprender a crearlos y ejecutarlos.

## ¿Qué es un guión por lotes?

Como ya intuyes **un guión o archivo por lotes es un archivo de texto cuyas líneas son comandos que se ejecutan secuencialmente**, te van a servir para agrupar órdenes y la extensión de este tipo de archivos es .bat o bien .cmd. En determinados casos, este tipo de ficheros permite ahorrar tiempo, ya que, una vez grabado en disco, con sólo escribir su nombre se ejecutan las órdenes que incluye.

La extensión **BAT** es propia y distintiva de los ficheros por lotes: todos tienen que tenerla y ningún otro fichero que no sea un fichero por lotes debe tenerla. La extensión BAT explicita que se trata de una orden que se ejecuta por línea de comandos.

Para ejecutar un fichero por lotes simplemente tienes que escribir el nombre del fichero (con o sin extensión) en la terminal o pulsar sobre el fichero bat.

Entre las **características** de este tipo de ficheros, podemos destacar las siguientes:

- Todos llevan extensión BAT.
- Cada orden ocupa una línea de la lista.
- Podemos incluir comandos internos y externos del sistema operativo, comandos especiales de los ficheros BAT, e incluso pueden llamar a otros ficheros ejecutables, que pueden ser otros ficheros BAT.
- No se puede asignar como nombre de un fichero BAT el de un comando externo o interno, ya que entonces sólo se ejecutará éste, y no el fichero.
- Podemos interrumpir la ejecución de un fichero BAT pulsando simultáneamente Control + C o Control + BREAK.
- Una vez guardado en disco, sólo es preciso escribir su nombre, sin indicar extensión, para ejecutarlo.

Si en un directorio concreto existe un fichero llamado program.exe y otro llamado program.bat, siempre se ejecutará primero el de extensión BAT. Es decir, si queremos ejecutar cualquiera de ellos, tendremos que especificar además su extensión.

## Edición de textos.

Vas a crear ficheros BAT, para ello puedes utilizar cualquier editor de texto plano. Puedes utilizar el editor de MS-DOS, es decir, EDIT. En caso de que tu versión de Windows 7 sea de 64 bits, no estará incluido el comando edit, podrás hacer los ficheros por lotes con el Bloc de notas como editor de texto, o cualquier otro editor de texto plano, por ejemplo el NOTEPAD.

También podemos crear ficheros de texto mediante la orden COPY CON fichero, que copia en el fichero directamente los caracteres introducidos en el teclado. Para ello escribes en la consola:

- Copy con nombre\_del\_archivo\_por\_lotes.bat
- Y escribimos el contenido del archivo, una vez acabado.
- Pulsamos control+Z (^Z).

## Comandos de ficheros BATCH.

A parte de todos los comandos propios del sistema operativo, un fichero BATCH puede contener una serie de comandos especiales de control, mediante los cuales podemos controlar el flujo de ejecución del comando, así como poner comentarios, instrucciones condicionales, etc. Estos comandos se llaman comandos de procesamiento por lotes.

**ECHO** Este comando le vas a usar cada vez que quieras introducir o mostrar un mensaje que quieras ver cuando ejecutes un archivo por lotes y para activar o desactivar la presentación de los comandos en la pantalla.

- **ECHO.** Muestra en pantalla el estado actual de eco.
- **ECHO ON.** Activa el eco y se ven en pantalla todos los mensajes.
- **ECHO OFF.** Desactiva el eco y no se ven en pantalla los mensajes hasta que se vuelve a activar.
- **ECHO mensaje.** Muestra en pantalla el mensaje que deseamos, independientemente de que el eco esté activado.

Cuando hagas un archivo por lotes la primera línea que escribirás será: **@echo off**

Con esta primera línea desactivas el echo y así, cuando ejecutes un fichero por lotes, verás el resultado de las órdenes escritas sin ver el comando que las ha realizado. Para eliminar el eco de la propia orden **ECHO OFF** se antepone el símbolo arroba (@) al nombre de la orden. A veces, te va a ser útil activar el echo para buscar los posibles errores del archivo por lotes, para ello, bastará quitar esta línea del archivo. **También cuando quieras incluir una línea en blanco en el archivo escribimos: echo. (echo punto).**

## REM

Te preguntará que hacen las líneas que comienzan por **::(dobles dos puntos)**, lo que te permite incluir comentarios explicativos en un archivo de proceso por lotes. También para incluir comentarios internos vas a poder usar **REM** o bien al final de la línea **con // seguido del comentario.**

Te recomiendo incluir comentarios en los ficheros de procesos por lotes, con el fin de clarificar lo que haces con cada comando o conjunto de comandos.

## PAUSE

Cuando estás ejecutando un archivo de proceso por lotes, puede que necesites pausar el resultado para ello vas a poder usar el comando **PAUSE**. Este comando suspende temporalmente la ejecución



del archivo hasta que se presione cualquier tecla, a menos que se pulse Control + BREAK o Control + C, en cuyo caso se suspenderá la ejecución del comando.

### Ficheros básicos.

Vas a crear archivos por lotes, es decir, un programa simple, que contendrá en cada línea una orden o comentario, les escribirás con un editor de texto plano y luego le guardaremos con la extensión .bat, aunque también podemos usar la extensión .cmd, para que el sistema operativo sepa que es un fichero ejecutable.

El fichero batch ejecuta una línea tras otra siguiendo las instrucciones que nosotros le damos. Detalles que debes tener en cuenta:

1. Saber que es lo que quieres que haga el fichero.
2. Estructurar el archivo, si va a tener un menú por ejemplo y empezar con una explicación de lo que hará ese fichero y si quieres desactivar el eco comenzaremos el archivo con la línea @echo off.
3. Introducir mensajes de salida (con el comando echo).

En el siguiente ejemplo puedes ver un archivo por lotes creado desde la terminal con la opción copy con y el nombre del archivo, en vez de utilizar el editor de textos.

Este fichero se guarda en mi directorio activo, en mi caso en C:\documents and Settings\Usuario. Otro ejemplo sencillo:

Tienes que saber que tenemos distintas maneras de crear un archivo por lotes:

1. Con el editor edit. (En versiones anteriores)
2. Con el bloc de notas o cualquier editor de texto plano.
3. Con el comando COPY CON nombre\_del\_fichero, escribimos sobre la terminal y para acabar pulsamos ^Z (control+z).
4. Con redireccionamiento.

### Ficheros con parámetros.

Tienes que saber que **los parámetros son informaciones adicionales colocadas detrás del nombre de una orden, es decir recibidas por un archivo por lotes**. Quizá en ocasiones el usuario o usuaria desee crear un programa y ejecutarlo con diferentes conjuntos de datos. Estos parámetros se introducen desde el símbolo del sistema, después del nombre de fichero de procesamiento por lotes.

Dentro de un fichero BATCH es posible utilizar hasta nueve parámetros, que se identifican por los caracteres %n, donde n es un número comprendido entre 1 y 9. Es decir, dentro de un fichero BATCH podemos emplear los **siguientes parámetros: %1, %2, ... , %9**.

Si el fichero BATCH incluye un parámetro, éste será sustituido por el nombre introducido después del nombre del fichero por lotes.

Cada parámetro es posicional respecto de lo que introducimos tras el nombre del fichero de proceso por lotes. Si introducimos dos parámetros, %1 y %2, el fichero BATCH esperará a que se le introduzcan dos parámetros tras el nombre del fichero. Cada parámetro se introduce separado por un espacio en blanco. El primer parámetro se sustituirá por % 1; el segundo, por %2, y así sucesivamente, pero de forma fija en todo el proceso.

Un ejemplo sumamente sencillo sería el de un script que recibe un parámetro que luego mostrará por pantalla.

## Ficheros con variables.

Tienes que saber que **una variable de entorno es una zona de memoria en donde se almacena un valor**, el formato de este comando.

En cmd, las variables se definen así:

```
set nombreVariable=valor
```

Aunque después, cuando las quieras utilizar, escribirás:  
%nombreVariable%

/A->Se utiliza para guardar expresiones en variables. Estas expresiones son numéricas ya que después podremos hacer operaciones con ellos.

/P->Almacena cualquier tipo de cadena (string) dentro de una variable, para poder ser llamada posteriormente.

set también puede ser puesta sin parámetros

Por ejemplo, si escribes en la línea de comandos las siguientes líneas

```
set /p saludo=hola  
echo %saludo%
```

Verás que te muestra la cadena "hola":

Si quieres que un archivo por lotes recoja una variable desde el teclado (a la que llamamos entrada), debes utilizar el parámetro /p en el comando set.

```
set /p nombreVariable=valor
```

```
Ejemplo para introducir un número y se suma con el valor de la variable v  
@echo off  
set v=5
```

```
set /p t=introduce numero
set/a p=%v%+%t%
echo %p%
```

## Ficheros con la estructura IF.

La orden **IF** desvía el proceso de ejecución de un fichero por lotes según sea verdadera o falsa una condición predeterminada. Para ver las opciones de if, escribimos en la terminal

Supón que tienes el siguiente fichero por lotes:

```
@echo off
del %1
```

Si ahora lo haces funcionar introduciendo como parámetro un fichero inexistente, te aparecerá un mensaje de error. Este fichero le puedes mejorar si contiene las siguientes líneas:

```
@echo off
if not exist %1 echo ¡Es imposible borrar un fichero que no existe!
if exist %1 del %1
```

### > **HELP IF**

Prueba a cambiar el archivo por lotes anterior por el siguiente:

```
@echo off
if "%1"==" " echo Debe introducir uno o dos ficheros como parámetros.
if not "%1"==" " del %1
if not "%2"==" " del %2
```

Si comparamos con el signo doble igual tanto la variable como lo que se compara va entre paréntesis

La primera línea **If** avisa si no se ha introducido ningún parámetro. La segunda línea únicamente borrará el primer parámetro si se ha introducido. La tercera línea hace lo mismo pero con el segundo parámetro.

## Ficheros con la orden FOR.

Su **sintaxis** es:

```
FOR %%variable IN (conjunto) DO orden
```

Esta orden repite la orden especificada para cada valor del conjunto. Donde "conjunto" es una lista de nombres de ficheros. En ella, se pueden establecer varios nombres separados por espacios y también, utilizar comodines.

**Ejemplo:**

for %%I in (juan.txt maria.txt \*.dat) do type %%i.

La variable %%I va tomando cada uno de los valores del conjunto y se los envía a la orden Type. En este ejemplo se visualizan en pantalla los ficheros JUAN.TXT, MARIA.TXT y todos los que tengan extensión DAT.

## Ficheros con la orden CALL.

Se utiliza para llamar a un fichero por lotes desde el interior de otro. Su **sintaxis** es:  
CALL fichero [ParámetrosDelFichero]

Vas a ver un **ejemplo**: el fichero PRG1.BAT llama a PRG2.BAT. Cuando la ejecución de PRG2.BAT termina, continúa PRG1.BAT en la siguiente línea a Call.

### Ejecución:

```
C:\BATH>prg1
```

```
línea 1
```

```
línea 2
```

```
línea 3
```

```
línea 4
```

```
línea 5
```

```
C:\BATH
```

## Ficheros con estructura GOTO.

La orden GOTO desvía el proceso de ejecución de un fichero por lotes a otra parte del fichero, para ello necesitamos dos elementos: **goto etiqueta** y **:etiqueta**

Aquí tienes enlaces a páginas con pequeños archivos por lotes explicados línea a línea:

```
@echo off
echo 1. fecha
echo 2. version
set /p op= ¿que opcion 1 o 2?
if "%op%"=="1" goto version
if "%op%"=="2" goto fecha
:fecha
echo La fecha de tu equipo es
date /t
goto fin
:version
echo La versión de tu sistema operativo es
ver
goto fin
:fin
```

Como puedes observar del ejemplo anterior, todas las etiquetas deben de comenzar con dos puntos. Tienes que tener cuidado con la manera de nombrar las etiquetas ya que no puedes poner el nombre

de un comando. Puedes utilizar la orden GOTO junto con la orden IF para crear bloques de órdenes que se ejecutarán tan solo si la condición IF es VERDADERA.

Te recuerdo que para comparar en la estructura **IF** tenemos que usar **doble signo de igual**. Cuando se emplea la orden GOTO, en las etiquetas que contienen órdenes puedes colocar dentro de ellas otra orden GOTO que salte a otro bloque de órdenes (como en el ejemplo GOTO FIN).

## ORDEN START.

Inicia una ventana separada para ejecutar un programa o comando especificado.

Modificadores:

/m [inimizado] ejecuta el nuevo programa minimizado (en segundo plano).  
/MAX [imizado] ejecuta el nuevo programa maximizado (en primer plano).  
/r [estaurado] ejecuta el nuevo programa restaurado (en primer plano).  
/w [ait] no vuelve hasta que se cierre el otro programa.  
/b crea la tarea en una nueva ventana

Ej:

start /MAX calc ‘ ejecuta la calculadora maximizada y continúa con la siguiente orden  
start /w notepad ‘ abre el bloc de notas y no continúa con la siguiente orden hasta que no se cierre  
start /w iexplore ‘ cuando se cierra el bloc de notas abre el explorador de internet

No vuelve el control al sistema hasta que no se cierran los programas, en caso de querer volver hay que añadir una orden **exit**.

## Errorlevel

Después de ejecutar algunas órdenes, dependiendo de cómo haya sido el resultado de su ejecución devuelve un código (errorlevel) que se chequea para comprobar cómo ha sido el resultado de esa ejecución. Se chequea comenzando por el nº más alto.

Vamos a crear un fichero por lotes para copiar los ficheros de la unidad c: a la c: e informe del resultado de la copia.

```
@echo off
rem Programa: COPIA-AB.BAT
xcopy c:\*.txt d:\
if errorlevel 1 goto Error
if errorlevel 0 echo ¡La copia fue correcta!
goto Final

:Error
echo Se produjo un error durante la copia

:Final
```

## Uso del comando AT

AT El comando AT programa la ejecución de comandos y programas en un equipo a una hora y fecha especificadas. El comando aún se encuentra disponible por cuestiones de compatibilidad, pero ha sido extendido en el comando SCHEDULETASKS que permite opciones más avanzadas. No obstante es posible emplearlo para la programación de tareas sencillas.

El servicio de programación debe estar en ejecución para utilizar el comando AT.

Sintaxis.

AT hh:mm ID [/EVERY:dia | /NEXT:dia] comando

Opciones.

- *ID Identidad que asignarle a la tarea, cualquier número.*
- *hh:mm Hora a la que se debe ejecutar el comando (hora:minutos en formato de 24 horas)*
- */every:dia Ejecuta el comando el, o los días especificados, las iniciales de los días utilizados tienen que corresponder a los días en el idioma ingles.*
- */next:dia Realiza la próxima ejecución del comando el día que se especifique*
- */delete Elimina tareas programadas, si no se especifica la ID se eliminan todas.*

Ejecutar como administrado

Ejemplos prácticos del uso del comando AT

Muestra todas las tareas programadas así como su ID (identidad)

AT

A la 1 de la madrugada inicia el navegador y conecta el equipo a Google.com.

AT 01:00 cmd /c start http://google.com

Realiza un respaldo, a las 8 de la mañana copia todos los archivos de la carpeta Mis Documentos en la carpeta Backup situada en el disco C.

AT 08:00 cmd /c copy %USERPROFILE%\Documents\\*. \* C:\

Todos los días ejecuta a las 11.30 de la mañana el mismo batch llamado copia\_diaria.cmd.

AT 11:30 /EVERY:m,t,w,th,f,s,su c:\backups\copia\_diaria.cmd

Todos los viernes a las 2.00 pm ejecuta el batch copia\_semanal.cmd.

AT 14:00 /EVERY:f c:\backups\copia\_semanal.cmd

Cancela la tarea de ID 8

AT 8 /delete

Cancela todas las tareas

AT /delete

## Comando Shutdown

Uso: shutdown [/i | /l | /s | /r | /g | /a | /p | /h | /e] [/f]

[/m \\equipo][/t xxx][/d [p|u:]xx:yy [/c "comentario"]]

Sin argumentos Muestra la ayuda. Es lo mismo que escribir /?.

/? Muestra la ayuda. Es lo mismo que no especificar ninguna opción.

/i Muestra la interfaz grafica de usuario (GUI).  
Debe ser la primera opción.

/l Cierra la sesión. No se puede utilizar con las opciones /m o /d.

/s Apaga el equipo.

/r Apaga y reinicia el equipo.

/g Cierra y reinicia el equipo. Después de reiniciar el sistema, reinicie las aplicaciones registradas.

/a Anula el apagado del sistema.  
Solo se puede usar durante el período de tiempo de espera.

/p Apaga el equipo local sin tiempo de espera ni advertencia.  
Se puede usar con las opciones /d y /f.

/h Hiberna el equipo local.  
Se puede usar con la opción /f.

/e Documenta la razón del cierre inesperado de un equipo.

/m \\equipo Especifica el equipo de destino.

/t xxx Establece el período de tiempo de espera antes del cierre en xxx segundos.

Quiero hacer un archivo en el escritorio para apagar el PC con el comando  
 "shutdown -s -t 0" ¿Cual sería el texto completo?  
 "shutdown -r -t 00"

## **Variables predefinidas**

Nosotros no podemos definir todos los nombres de variables que queramos, ya que algunos nombres ya vienen cogidos por nuestro sistema operativo y tienen su función correspondiente. En esta parte vamos a nombrar algunos de ellos, ya que puede que en algunos programas requiráis una lectura de los mismos. Estas variables son:

%ALLUSERSPROFILE% => Esta variable devuelve la localización del perfil de todos los usuarios.

%APPDATA% => Devuelve el lugar donde las aplicaciones guardan los datos por defecto (Normalmente la carpeta de Archivos de Programa).

%CD% => Devuelve el directorio en el que estás en ese momento (se explicará más adelante con sus comandos para cambiarse de directorio).

%CMDCMDLINE% => Muestra el comando exacto empleado para acceder al intérprete de comandos (cmd.exe).

%CMDEXTVERSION% => Devuelve la extensión de nuestro intérprete de comandos.

%COMPUTERNAME% => Devuelve el nombre del equipo.

%COMSPEC% => Devuelve la ruta de la shell de comandos.

%DATE% => Devuelve la fecha actual.

%ERRORLEVEL% => Devuelve el código de error del último comando ejecutado.

%HOMEDRIVE% => Devuelve la unidad en la que está el directorio en el que estás actualmente.

%HOMEPATH% => Devuelve la ruta completa a dicho directorio.

%LOGONSERVER% => Devuelve el nombre de nuestro servidor.

%NUMBER\_OF\_PROCESSORS% => Devuelve el número de procesadores instalados en el equipo.

%OS% => Devuelve nuestro sistema operativo, con la excepción de Windows 2000 y XP que lo devuelven como Windows\_NT.

%PATH% => Devuelve la ruta a la carpeta de los ejecutables más importantes del sistema.

%PATHEXT% => Devuelve las extensiones de archivos que nuestro sistema considera ejecutables.

%PROCESSOR\_ARCHITECTURE% => Devuelve la arquitectura del procesador.

%PROCESSOR\_IDENTIFIER% => Devuelve la descripción del procesador.

%PROCESSOR\_LEVEL% => Devuelve el número de modelo de procesador.

%PROCESSOR\_REVISION% => Devuelve el número de revisión del procesador.

%PROGRAMFILES% => Devuelve la carpeta donde se guardan los programas (normalmente Archivos de Programa).

%RANDOM% => Devuelve un número al azar entre 0 y 32767.



%SYSTEMDRIVE% => Devuelve la unidad que contiene el directorio raíz del sistema.

%SYSTEMROOT% => Devuelve la carpeta de administración, que suele ser C://Windows

%TEMP% => Devuelve el directorio donde están los archivos temporales.

%TMP% => Igual que el anterior.

%TIME% => Devuelve la hora actual.

%USERNAME% => Devuelve el nombre del usuario actual.

%USERPROFILE% => Devuelve la ruta del directorio donde están los archivos del usuario actual.

%WINDIR% => Devuelve la ruta de la carpeta del sistema operativo

De esta forma nosotros podemos hacer por ejemplo un programa que nos muestre la hora actual, la fecha y el usuario que la ha visto de la siguiente forma:

```
@echo off
title Hora actual
echo El usuario %USERNAME% ha solicitado ver la hora
pause
echo La fecha es %DATE% y son las %TIME%
pause>nul
exit
```

Ejemplo que muestra el nombre de usuario y la fecha y hora

```
@echo off
title Hora actual
echo El usuario %USERNAME% ha solicitado ver la hora
pause
echo La fecha es %DATE% y son las %TIME%
pause>nul
exit
```

# Listado de Comandos para Windows 7

Microsoft Windows 7 también se puede administrar desde la línea de comandos. Para acceder a la línea de comandos de Windows 7, se pulsa en el Menú Inicio y en el campo de Buscar Programas y Archivos (o la tecla de Windows + r) se escribe el comando **cmd** y se pulsa enter. De esta forma se abre la consola del sistema donde se podrá entrar los comandos.

A continuación se muestra una lista con algunos de los comandos de Windows 7 ordenados alfabéticamente: (esta misma lista aparece usando el comando help)

**ASSOC** - Muestra o modifica las asociaciones de las extensiones de archivos.  
**ATTRIB** - Muestra o cambia los atributos del archivo.  
**BREAK** - Establece o elimina la comprobación extendida de Ctrl+C.  
**BCDEDIT** - Establece propiedades en la base de datos de arranque para controlar la carga del arranque.  
**CACLS** - Muestra o modifica las listas de control de acceso (ACLs) de archivos.  
**CALL** - Llama a un programa por lotes desde otro.  
**CD** - Muestra el nombre del directorio actual o cambia a otro directorio.  
**CHCP** - Muestra o establece el número de página de códigos activa.  
**CHDIR** - Muestra el nombre del directorio actual o cambia a otro directorio.  
**CHKDSK** - Comprueba un disco y muestra un informe de su estado.  
**CHKNTFS** - Muestra o modifica la comprobación de disco al arrancar.  
**CLS** - Borra la pantalla.  
**CMD** - Inicia una nueva instancia del intérprete de comandos de Windows  
**COLOR** - Establece los colores de primer plano y fondo predeterminados de la consola.  
**COMP** - Compara el contenido de dos archivos o un conjunto de archivos.  
**COMPACT** - Muestra o cambia el estado de compresión de archivos en particiones NTFS.  
**CONVERT** - Convierte volúmenes FAT a volúmenes NTFS. No puede convertir la unidad actual.  
**COPY** - Copia uno o más archivos en otra ubicación  
**DATE** - Muestra o establece la fecha.  
**DEL** - Elimina uno o más archivos.  
**DIR** - Muestra una lista de archivos y subdirectorios en un directorio.  
**DISKCOMP** - Compara el contenido de dos disquetes.  
**DISKCOPY** - Copia el contenido de un disquete en otro.  
**DISKPART** - Muestra o configura las propiedades de partición de disco.  
**DOSKEY** - Edita líneas de comando, memoriza comandos de Windows y crea macros.  
**DRIVERQUERY** - Muestra el estado y las propiedades actuales del controlador de dispositivo.  
**ECHO** - Muestra mensajes, o activa y desactiva el eco.  
**ENDLOCAL** - Termina la búsqueda de variables de entorno del archivo por lotes.  
**ERASE** - Elimina uno o más archivos.  
**EXIT** - Sale del programa CMD.  
**EXE** (interfaz de comandos).  
**FC** - File Compare - Compara dos archivos o conjunto de archivos y muestra las diferencias entre ellos.

**FIND** - Busca una cadena de texto en uno o más archivos.

**FINDSTR** - Busca cadenas de texto en archivos.

**FOR** - Ejecuta un comando para cada archivo en un conjunto de archivos.

**FORMAT** - Formatea un disco para usarse con Windows.

**FSUTIL** - Muestra o configura las propiedades de sistema de archivos.

**FTYPE** - Muestra o modifica los tipos de archivo usados en una asociación de extensión de archivo.

**GOTO** - Direcciona el intérprete de comandos de Windows a una línea en un programa por lotes.

**GPRESULT** - Muestra información de directivas de grupo por equipo o usuario.

**GRAFTABL** - Permite a Windows mostrar un juego de caracteres extendidos en modo gráfico.

**HELP** - Proporciona información de ayuda para los comandos de Windows.

**ICACLS** - Muestra, modifica, hace copias de seguridad o restaura listas de control de acceso (ACL) para archivos y directorios.

**IF** - Ejecuta procesos condicionales en programas por lotes.

**LABEL** - Crea, cambia o elimina la etiqueta del volumen de un disco.

**MD** - Crea un directorio.

**MKDIR** - Crea un directorio.

**MKLINK** - Crea vínculos simbólicos y vínculos físicos.

**MODE** - Configura un dispositivo de sistema.

**MORE** - Muestra la información pantalla por pantalla.

**MOVE** - Mueve uno o más archivos de un directorio a otro en la misma unidad.

**NETSH** - Network Shell - Permite manejar la configuración de red del sistema.

**OPENFILES** - Muestra archivos compartidos abiertos por usuarios remotos.

**PATH** - Muestra o establece una ruta de búsqueda para archivos ejecutables.

**PAUSE** - Suspende el proceso de un archivo por lotes y muestra un mensaje.

**POPD** - Restaura el valor anterior del directorio actual guardado por PUSH.D.

**PRINT** - Imprime un archivo de texto.

**PROMPT** - Cambia el símbolo de comandos de Windows.

**PUSH.D** - Guarda el directorio actual y después lo cambia.

**RD** - Quita un directorio.

**RECOVER** - Recupera la información legible de un disco dañado o defectuoso.

**REM** - Marca comentarios en archivos por lotes o CONFIG.

**SYS** El comando SYS se utiliza para copiar los archivos de sistema de un disco a otro. El PC necesita estos archivos para iniciarse.

**REN** - Cambia el nombre de uno o más archivos.

**RENAME** - Cambia el nombre de uno o más archivos.

**REPLACE** - Reemplaza archivos.

**RMDIR** - Quita un directorio.

**ROBOCOPY** - Utilidad avanzada para copiar árboles de archivos y directorios.

**SET** - Muestra, establece o quita variables de entorno de Windows.

**SETLOCAL** - Comienza la sección de cambios locales de entorno en un archivo por lotes.

**SC** - Muestra o configura servicios (procesos en segundo plano).

**SCHTASKS** - Programa comandos y programas para ejecutarse en un equipo.

**SHIFT** - Cambia posición de modificadores reemplazables en archivos por lotes.

**SHUTDOWN** - Permite el apagado local o remoto de un equipo.

**SORT** - Ordena la salida.

**START** - Inicia otra ventana para ejecutar un programa o comando.

**SUBST** - Asocia una ruta de acceso con una letra de unidad.

**SYSTEMINFO** - Muestra las propiedades y la configuración específicas del equipo.

**TASKLIST** - Muestra todas las tareas en ejecución, incluidos los servicios.

**TASKKILL** - Termina o interrumpe un proceso o aplicación que se está ejecutando.

**TIME** - Muestra o establece la hora del sistema.

**TITLE** - Establece el título de la ventana de una sesión de CMD.

**EXE**

**TREE** - Muestra gráficamente la estructura de directorios de una unidad o ruta de acceso.

**TYPE** - Muestra el contenido de un archivo de texto.

**VER** - Muestra la versión de Windows.

**VERIFY** - Comunica a Windows si debe comprobar que los archivos se escriben de forma correcta en un disco.

**VOL** - Muestra la etiqueta del volumen y el número de serie del disco.

**XCOPY** - Copia archivos y árboles de directorios.

**WEVTUTIL** - Se gestionan los logs de los sistemas Windows, configuración, borrado, almacenamiento

**WMIC** - Muestra información de WMI (Windows Management Instrumentation) en el shell de comandos interactivo.