

## Comandos DOS

Vamos a ver algunos comandos del DOS pero sin llegar a aburrirnos y ejecutar uno por uno desde la A a la Z. Lo intentaremos hacer ameno con algo de programación, más bien algo de *scripting* dividido en los siguientes apartados:

1. Ficheros por lotes
2. Aplicación Práctica
3. Redireccionamiento de entrada y salida estándar.
4. Prácticas de Ficheros Batch
5. Prácticas de Redireccionamiento y filtros
6. ANEXO 1: Lista de comandos de Consola presentes en la Versión Windows 2000

## Indice de Contenidos

Comandos DOS.....	1
Indice de Contenidos .....	2
1. Ficheros por lotes.....	3
2. Aplicación Práctica .....	6
3. Redireccionamiento de entrada y salida estándar.....	7
Resumen Redireccionamiento:.....	9
4. Prácticas de Ficheros Batch.....	10
4.1 Realizar un script buscar.bat al que se le pase como parametro un nombre de archivo con comodines (*/*) y localice en disco los que se le parezcan.....	10
4.2 Realizar un script que implemente una especie de papelera de reciclaje en la carpeta c:\basura .....	10
4.3 Vamos a hacer ahora uno complicado, este programa servirá para copiar un directorio que no cabe en un único disquete, de forma que vaya pidiendo los sucesivos.....	11
5. Prácticas de Redireccionamiento y filtros .....	12
5.0 Crear un fichero llamado AMIGOS1.TXT que contenga los siguientes datos.....	12
5.1 Conocer todos los amigos de Córdoba. ....	13
5.2 Conocer el número de teléfono de Cristina. ....	13
5.3 Ordenar alfabéticamente el fichero basándonos en la ciudad. ....	13
5.4 Ordenar alfabéticamente el fichero basándonos en el apellido.....	13
5.5 Crear otro fichero llamado AMIGOS2.TXT que contenga lo siguiente: .....	13
5.6 Unir ambos ficheros en un tercero llamado AMIGOS3.TXT .....	13
5.7 Ordenar alfabéticamente AMIGOS3.TXT por el nombre y escribir el resultado en un nuevo fichero llamado AMIGOS4.TXT. ....	13
5.8 Visualizar por pantalla AMIGOS4.TXT. Si tienes impresora, imprímelo. ....	13
5.9 Borrar los cuatro ficheros creados (pero sólo esos 4!) .....	13
5.10 Soluciones a los ejercicios propuestos.....	13
ANEXO 1: Lista de comandos de Consola presentes en la Versión Windows 2000 .....	15

## 1. Ficheros por lotes

Los ficheros por Lotes llamados también Batch Files en DOS tiene la extensión “.bat”. Y no son más que ficheros ASCII que podemos editar con el Notepad y permiten la ejecución secuencial de las instrucciones, más bien de los comandos, que llevan dentro.

Todas las órdenes admitidas en el símbolo del sistema o consola pueden introducirse también en un fichero por lotes. Además, existen una serie de órdenes diseñadas específicamente para estos ficheros que permiten aunque limitadamente cierta programabilidad de los mismos y control del flujo:

Orden	Breve descripción
<b>Echo</b>	Controla el eco de las órdenes y visualiza mensajes.
<b>Rem</b>	Introduce comentarios.
<b>Pause</b>	Detiene temporalmente el desarrollo de un programa.
<b>Goto</b>	Desvía incondicionalmente el desarrollo de un programa.
<b>If</b>	Desvía condicionalmente el desarrollo de un programa.
<b>For</b>	Repite una misma orden en un conjunto de ficheros.
<b>Call</b>	Llama a un fichero por lotes desde otro.

La siguiente tabla muestra aquellos símbolos empleados exclusivamente en los ficheros por lotes:

Símbolo	Significado
<b>:etiqueta</b>	Nombre de una etiqueta.
<b>%número</b>	Parámetro del fichero por lotes.
<b>%variable%</b>	Variable del entorno.
<b>%%variable</b>	Variable de la orden For.
<b>@Echo off</b>	“Predesactiva” el Echo
<b>Errorlevel</b>	Resultado del último comando

Para que os familiariceis rápidamente vamos a hacer unos scripts muy sencillos de los que debéis entender lo que hacen, y donde se ponen en juego todas las posibilidades:

- Formatear un disquete y comprobar el resultado devuelto al Sistema Operativo
- Chekear la integridad de un disquete (este lo haceis vosotros modificando levemente el primero)
- Envenenamiento por mensajes
- Lectura de parámetros desde la línea de ordenes
- Llamar a otros scripts desde el mio

Los scripts:

### 1. Formatear disquete

```
:INICIO
@echo off
Rem Script que formatea un disquete
format a:
if not errorlevel 1 goto BIEN
:MAL
echo Ocurrio un Error durante el formateo
goto FIN
:BIEN
echo Formateo OK
:FIN
pause
```

### 2. Chequear disquete

Sustituir el comando « FORMAT » por « CHKDSK » y poner el comentario (Rem) y mensajes (Echo) adecuados.

### 3. Envenenamiento por mensajes

Cuidado este script puede hacer que la maquina destino deje de funcionar. Es un claro ejemplo de fuerza bruta.

Consiste en utilizar el comando NET SEND para enviar un numero de veces seguidas el mismo mensaje a un equipo de la Red. Antes de continuar ejecutar lo siguiente para averiguar la sintaxis del comando:

**C:\>NET SEND**

NET SEND {nombre | \* | /DOMAIN[:nombre] | /USERS} mensaje

Si escribes solo « NET » se listará el conjunto de utilidades disponibles:

NET [ ACCOUNTS | COMPUTER | CONFIG | CONTINUE | FILE | GROUP | HELP |  
HELPMMSG | LOCALGROUP | NAME | PAUSE | PRINT | SEND | SESSION |  
SHARE | START | STATISTICS | STOP | TIME | USE | USER | VIEW ]

Así pues podeis averiguar la sintaxis para saber como se utiliza “NET SHARE” escribiendo la orden:

**C:\>NET SHARE /?**

A lo que vamos, el script que vamos a utilizar contendrá como habeis adivinado el comando **FOR** cuya sintaxis es la siguiente, notar la diferencia entre **%%variable** y **%variable**:

- En un fichero batch:  
FOR %%variable IN (set) DO command [command-parameters]
- Desde el command prompt:  
FOR %variable IN (set) DO command [command-parameters]

**%%variable or %variable**

Representa una variable. El FOR reemplaza %%variable (or %variable) en cada iteración del bucle, por los valores incluidos entre los parentesis.

**(set)**

Especifica el conjunto de valores para la variable, y puede interpretarse como uno o más ficheros o cadenas de texto que quieres procesar.

**command**

Especifica el comando en cuestion que se ejecutara para cada valor de la variable.

**command-parameters**

Es posible pasar parametros al comando utilizado si los tiene o son necesarios.

El Script, sólo 3 líneas de las que podemos ahorrarnos 2:

```
@ECHO OFF
for %%f in (1 2 3 4 5 6 7 8 9 10) do NET SEND IES-EQUIPO1 HOLA
PAUSE
```

Donde pone IES-EQUIPO1 poner el nombre de un equipo de los que veis en vuestro Entorno de Red. Podeis verlos así también:

C:\>NET VIEW

Y os recuerdo cuidado con este script. Para evitar que os molesten podeis acceder a los “Servicios” de la máquina y desactivar el de “mensajería”.

#### 4. Lectura de parámetros desde la línea de ordenes

*Los parámetros son informaciones adicionales colocadas detrás del nombre de una orden.* Ya que la mayoría de las órdenes de DOS y de cualquier Sistema Operativo admiten parámetros, a la hora de programar Batch también se permite el uso de parámetros. En DOS en particular podemos pasarle a un Script hasta 9 parámetros. Después desde el script podemos acceder al tercer parametro por ejemplo escribiendo %3.

Vamos a verlo en acción, para ello haremos un script que copie dos ficheros a la unidad de disco, pasando como parámetros dos ficheros. La llamada a mi script desde la línea de comandos seria algo así:

C:\>copiarA listado.txt entrada.dat

Y el script quedaría así:

```
@echo off
rem Programa: COPIARA.BAT
echo Se están copiando los ficheros %1 y %2 a la unidad A:
copy %1 a:\
copy %2 a:\
PAUSE
```

#### 5. Llamar a otros scripts desde el mio

Para probar esta funcionalidad lo que podriamos hacer es una especie de menu. Sera sencillo de 2 opciones tan solo. La primera opción pondría Etiqueta (Label) a un Disquete en A:, y la segunda realizaría un chequeo del mismo.

La primera opcion será un fichero bat con estas líneas, y que llamaremos **etiqueta.bat**:

```
@echo off
rem Programa: ETIQUETA.BAT
LABEL A:
```

La segunda la teneis ya en el script (2) y vamos a suponer que le llamamos **chequear.bat**

```
@echo off
rem Programa: CHEQUEAR.BAT
CHKDSK A:
```

Y nuestro fichero de menu seria el tercer y ultimo script al que llamaremos **menu.bat**:

```
INICIO:
    @ECHO OFF
    REM MENU DE OPCIONES
    CLS
    if "%1"=="1" goto OPCION1
    if "%1"=="2" goto OPCION2
    ECHO OPCION NO VALIDA
    ECHO  MENU OPCION
    ECHO  DONDE OPCION PUEDE SER:
    ECHO          1: ETIQUETA
    ECHO          2: CHEKEO
    GOTO FIN
:OPCION1
    CALL ETIQUETA.BAT
    goto FIN
:OPCION2
    CALL checkA.BAT
:FIN
    ECHO PULSA UNA TECLA
    pause>null
```

Que para ejecutarse deberiamos escribir estas dos ordenes:

```
C:\>menu 1
Con lo que pondria etiqueta a la unidad A:
C:\>menu 2
Con lo que chekearia la unidad A:
```

En este último script he utilizado la orden PAUSE > NUL, te das cuenta de lo que hace, si no lo pillas, después de ver el punto 3 lo haras.

## 2. Aplicación Práctica

**Objetivo:** Vamos a “mapear” una carpeta de nuestro disco para acceder a ella desde una letra de unidad (W: de Work estaria bien). Pero queremos que el mapeo sea permanente, es decir, que cada vez que yo inicie sesion en la maquina tenga disponible la unidad para ponerme a trabajar.

**Comando:** El comando que vais a estudiar es el “SUBS”

Asocia una ruta de acceso con una letra de unidad.

```
SUBST [unidad1: [unidad2:]ruta]
SUBST unidad1: /D
Unidad1:      Especifica la unidad virtual a la que se asigna la ruta.
[Unidad2:]    Ruta Unidad física y ruta de acceso que se desea asignar a una unidad virtual.
/D           Elimina una unidad sustituida (virtual).
Escriba SUBST sin parámetros para ver la lista de los dispositivos virtuales.
```

Si por ejemplo queremos que la unidad **W:** se mapee sobre el directorio “C:\Documents and Settings\Usuario1\Mis documentos\jose blanco” lo que va en negrita será diferente para cada uno, la orden a lanzar sería:

```
C:\>subst w: "C:\Documents and Settings\Usuario1\Mis documentos\jose blanco"
```

En modo consola a lo mejor no os funcionará. Hacer un nuevo archivo de texto con el bloc de notas y escribir además de la línea del comando en sí, el comando “PAUSE” debajo para realizar una pausa tras la ejecución del SUBST y ver si funcionó.

Cuando funcione bien el script y tengáis unidad W: hacer otro script logoff.bat que borre el mapeado de W: sobre vuestra carpeta de trabajo.

Llegados a este punto podemos pasar a la práctica que veníamos comentando, que la unidad W: este disponible siempre que arranquemos.

**Practica:** Esto como todo se puede hacer de varias formas, veamos algunas de ellas:

**Primera Técnica:** Colocar un acceso directo o el propio archivo por lotes (.bat) en la carpeta Inicio de nuestro Perfil (ya sea alumno o el personalizado que tengáis. Una forma fácil de acceder a la carpeta Inicio es hacer botón derecho sobre el botón de inicio y navegar hasta encontrar la rama. Dentro de la Carpeta Programas entre otras está la de inicio. Normalmente en “C:\Documents and Settings\Usuario\Menú Inicio\Programas\Inicio”. Pues ahí dentro dejar el script o un acceso directo y reiniciar o simplemente cerrar la sesión. Para colocar el script copiar desde origen y pegar aquí dentro, para hacer un acceso directo al mismo, botón derecho sobre el destino (Inicio) y NUEVO/ACCESO DIRECTO. Localizáis el archivo logon.bat en disco y siguiente, podéis dejar el nombre por defecto y ya lo tendríais.

**Segunda Técnica:** Colocar una entrada en el registro. Esta es más peligrosa pero si se siguen los pasos no tendréis ningún problema. Para abrir el editor del registro ejecutar “REGEDIT” sin las comillas, desde INICIO/EJECUTAR. A cada uno se os abrirá en una rama, depende de la última vez que se accedió si es que se accedió alguna vez. Debeis navegar hasta la ruta: **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run**

En esa ruta están colocados todos los programas que necesita el sistema operativo o algún dispositivo para funcionar, aunque a veces hay otras entradas menos necesarias. Para quitar cosas de ahí conviene saber lo que se está haciendo y conocer que hace cada ejecutable o librería (dll) antes de estropear algo. Como todo tiene solución, antes de empezar a jugar vamos a hacer una copia de seguridad de esta rama. Para ello botón derecho sobre la carpeta run a la izquierda, y “EXPORTAR”, guardarla con un nombre como rama\_run, y en una carpeta que controleis vosotros. Generará en disco un archivo de extensión .reg, rama\_run.reg, cuidado si le hacéis doble clic ejecutaría el script que lleva dentro y daría de alta las claves que contenga el mismo. Lo que podéis hacer es mirarle las tripas, con el botón derecho podéis “EDITAR”, no es más que un fichero ASCII fácilmente modificable si se sabe.

Para seguir con lo nuestro, botón derecho sobre la rama Run y seleccionar esta vez NUEVO/VALOR ALFA. Le poneis como nombre “logon” sin las comillas y hacéis doble clic sobre el. Suponiendo que hemos guardado el script en la raíz del disco C: en el valor escribimos “c:\logon.bat” sin las comillas. Y cerráis la sesión para volverla a abrir.

### 3. Redireccionamiento de entrada y salida estándar.

**Introducción:** Cuando escribimos algo en la consola de comandos se reproduce en pantalla a la medida que tecleamos (es el llamado **echo** en contexto de sistemas informáticos), esto ocurre por que por defecto la **Salida Estándar** es la **pantalla**. Por otro lado cuando estamos en **modo consola** el sistema operativo recibe los comandos que introducimos por **teclado**, que por defecto es la **Entrada Estándar**. Bien pues esto es así por defecto, pero puede cambiar, y el mayor fruto a corto plazo que podemos obtener es por ejemplo redireccionar la salida de un comando a un fichero de texto de la siguiente manera:

```
C:\>DIR /W > listado.txt
```

Con lo cual el comando DIR /W que lista el contenido de la carpeta actual con cierto formato en lugar de salir por pantalla a salido por otro Stream en este caso de tipo File (File Stream)

que hemos decidido llamar listado.txt. Si veis el contenido del archivo lo vereis vosotros mismos.

**Cómo:** Aquí lo que hemos utilizado es un **operador >** que indica al sistema operativo que **redireccione la salida al dispositivo indicado**, en este caso un fichero, pero podría haber sido una impresora indicando LPT1 si está conectada al primer puerto paralelo , o bien PRN que es un dispositivo que indica la impresora por defecto en el sistema. El nombre especial de dispositivo para pantalla es CON (CONsola).

Con el **operador <** somos capaces de **redireccionar la entrada estandar a un dispositivo** o Stream distinto del teclado. Aquí el ejemplo no es tan trivial, pero es igual de facil de probar. Crear un archivo de texto ASCII y llamarlo entrada.dat, escribir en el lo siguiente:

midisco

Guardarlo, cerrarlo y desde consola ejecutar lo siguiente:

```
C:\>LABEL A: < ENTRADA.DAT
```

Esto lo que hara es que cuando el comando pregunte el nombre de etiqueta para el disquete en lugar de esperarlo por teclado lo lee del archivo. El archivo Entrada.dat no es más que un “archivo de parámetros”.

Por ultimo el **operador >>** (mayormayor) permite concatenar al dispositivo el resultado de la salida estandar conservando lo que ya tenia. Para comprobarlo ejecutar:

```
C:\>DIR /W >> listado.txt
```

Y vereis como al final del primer listado concatena el segundo.

Por último en esto de los redireccionamientos existe una cosa que se llama Pipe o Tuberia, que en Linux se utiliza para casi todo, el tema del Tunelado consiste en hacer que la salida de un comando se la coma otro comando, y la sintaxis general es:

```
C:\>comando1|comando2
```

Donde comando1 es el primero que se ejecuta y todo lo que suelta se lo traga comando2. Ambos pueden tener por su puesto su ristra de parametros y opciones.

Un ejemplo:

```
C:\>DIR|FIND "MENU"
27/01/2007 13:08          325 MENU.BAT
C:\>
```

El DIR simplemente habria sacado el listado de archivos y carpetas de la unidad C :, pero nosotros solo queremos aquellos archivos que se llamen “MENU”, esta seria una forma de obtenerlo filtrando la información que suelta DIR y sacando solo aquellas líneas que coincidan con el patrón de búsqueda. Esto es el tuneling.

Los filtros del dos son las órdenes More, Sort y Find. Estas tres órdenes externas se denominan filtros porque su misión es recibir una entrada, filtrarla y devolver una salida, aunque se puede jugar con ellos sin hacer filtrado:

```
C:\>SORT
```

Al ejecutar esa orden se queda la consola esperando entradas separadas por INTROS para finalizar con un CTRL-Z y un INTRO que graba la marca de fin de fichero EOF. Para después mostrarnos por pantalla la secuencia ordenada.



**More:** Recibe como entrada un conjunto de caracteres organizados en filas y devuelve los mismos caracteres insertando una pausa cada 23 líneas. En muchas ocasiones la información mostrada por una orden o el contenido de un fichero no cabe en una sola pantalla y los datos se desplazan a gran velocidad sin darnos tiempo a leerlos. La utilidad de la orden More se encuentra en estas ocasiones, deteniendo el desplazamiento de información en cada pantalla. Entonces, More muestra la siguiente línea:

-- Más --

Después de leer los datos en pantalla, podemos pulsar Enter para continuar o también Ctrl+Pausa para interrumpir.

**Sort:** Recibe como entrada un conjunto de caracteres organizados en filas y devuelve los mismos caracteres ordenando cada fila alfabéticamente. El uso más frecuente de este filtro es ordenar alfabéticamente el contenido de un fichero. Admite dos modificadores:

/R Ordena inversamente (de la `Z' a la `A' y del `9' al `0')

/+columna Indica la columna del carácter a partir del cual se comparará cada fila.

**Find:** Recibe como entrada un conjunto de caracteres organizados en filas y devuelve las filas que contengan una cadena determinada. La cadena de caracteres para buscar debe especificarse entre comillas (Mays+2) después de la orden. Se utiliza, a menudo, para buscar una cadena de caracteres en un fichero o en la salida de una orden. El filtro Find **distingue entre mayúsculas y minúsculas** mientras no le especifiquemos el modificador /I entre la orden y la cadena:

FIND "cadena" (cadena es diferente que CADENA)

FIND /I "cadena" (cadena es lo mismo que CADENA)

#### Resumen Redireccionamiento:

Operador