

Windows Scripting (part 1)

1.- La línia de comandes de Microsoft

La línia de comandes dels sistemes Microsoft Windows ha anat evolucionant des de les seves primeres versions de MS-DOS. Sovint associem la línia de comandes de Windows amb MS-DOS i són dos coses ben diferents. MS-DOS, com a sistema operatiu sense entorn gràfic està pràcticament obsolet i no s'utilitza (reemplaçat per les successives versions de Windows). No obstant, la línia de comandes forma part de totes les successives versions de Windows i mai ha deixat de tenir un paper actiu (al contrari, Microsoft cada cop ha anat potenciant-la més i més).

Existeixen dues versions de la interface de comandes dels sistemes Microsoft Windows:

- (1) la línia de comandes tradicional (COMMAND.COM i CMD.EXE), derivada dels sistemes MS-DOS
- (2) Windows Powershell, el qual va aparèixer a partir de Windows XP.

L'entorn Powershell permet l'execució de les mateixes comandes que la línia de comandes tradicional de Microsoft Windows i també altres comandes incorporades a Powershell (les quals no estan presents a la línia de comandes tradicional). Per tant, Powershell inclou un conjunt de comandes més ampli, el qual està basat en l'ús d'objectes de la plataforma .NET de Microsoft, de manera que gestiona objectes complexes com a entrada i sortida de les comandes mentre de gestionar cadenes de text.

A continuació anem a veure la línia de comandes tradicional.

2.- La línia de comandes tradicional de Windows

Tal com s'ha esmentat en l'apartat anterior, la línia de comandes tradicional dels sistemes Microsoft Windows té dues versions : COMMAND.COM i CMD.EXE. El CMD.EXE és present només en sistemes Windows 2000 i posteriors.

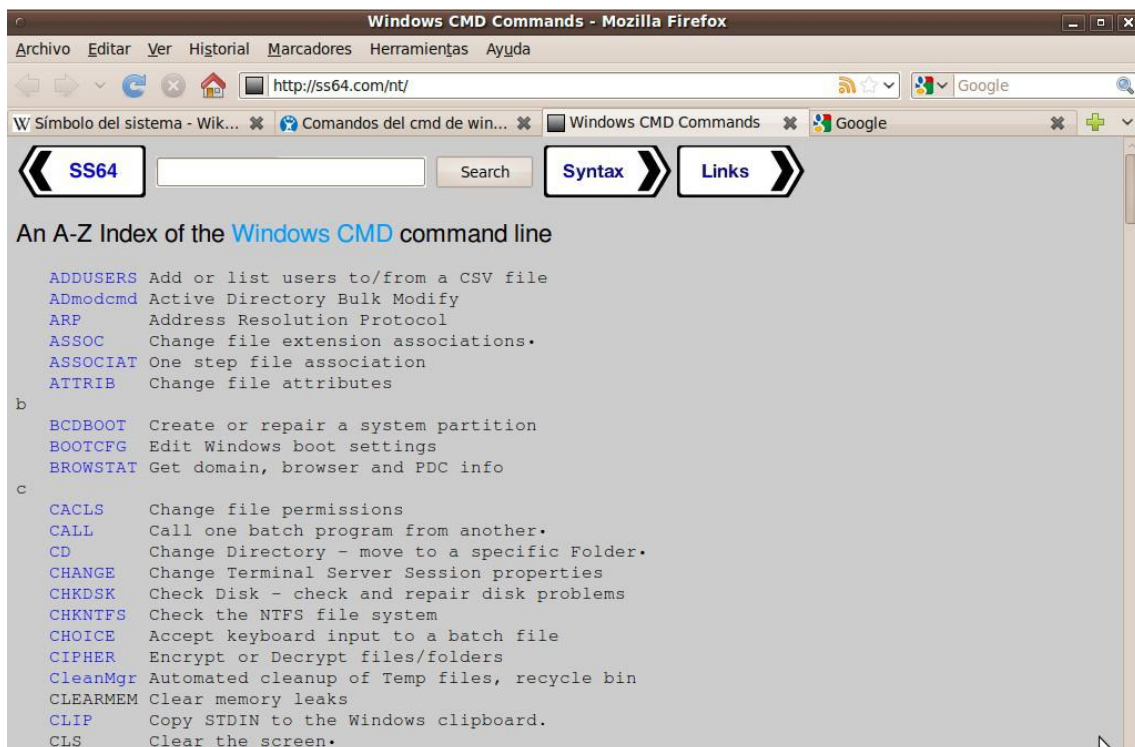
Aquestes dues versions són molt semblants, ja que els conjunts de comandes que accepten ambdúes són gairebé idèntics. La diferència principal radica en que COMMAND.COM està preparat per a funcionar sobre el sistema operatiu MS-DOS mentre que CMD.EXE és un programa que funciona en una finestra de Windows. Seguidament ens centrarem en el CMD.EXE, que és un entorn que s'instal·la per defecte en els sistemes Windows 2000/XP/2003/Vista.

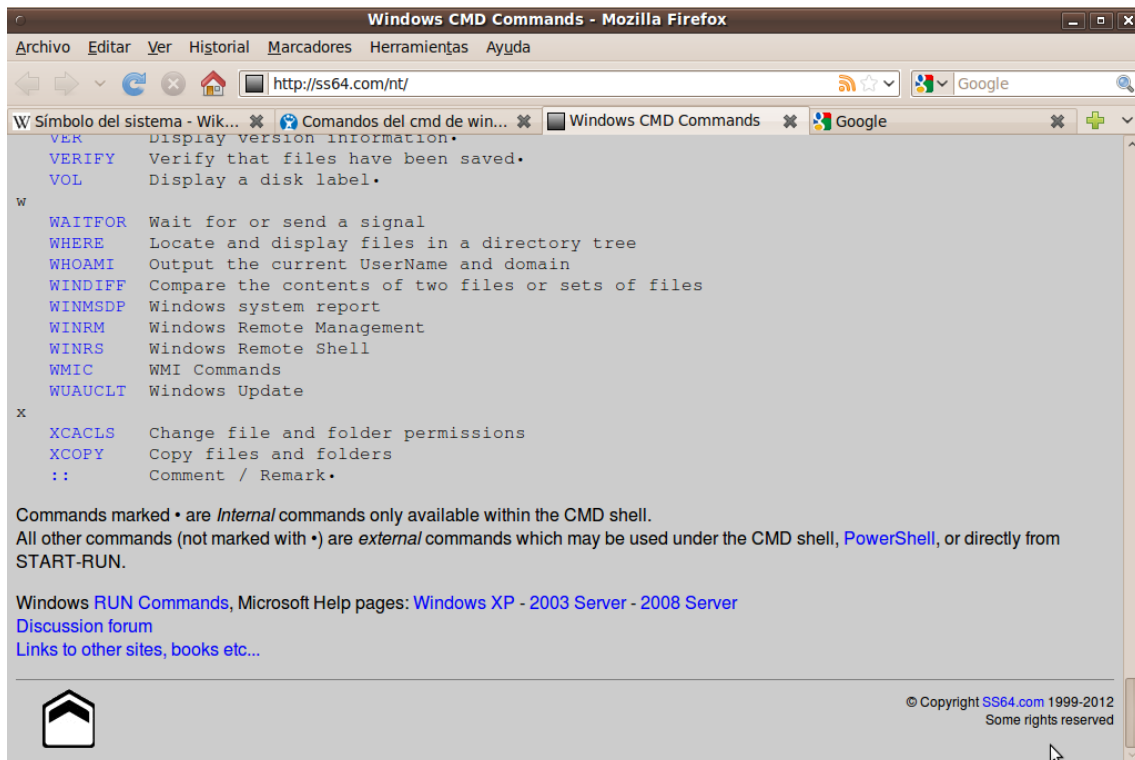
El símbol de sistema o CMD.EXE funciona escrivint comandes i obtenint resultats. Les comandes es poden introduir en majúscules o minúscules, ja que l'interpret no diferencia entre majúscules i minúscules. Les rutes absolutes o

relatives s'especifiquen utilitzant la barra invertida “\” per separar els noms de les carpetes.

S'anomenen comandes internes aquelles que estan emmagatzemades dins del propi intèrpret de comandes o el nucli del sistema operatiu i per altra banda hi ha les comandes externes que són aquelles que s'emmagatzemen en arxius executables independents.

A la web que veiem en les dues imatges següents trobem una bona referència de les comandes del cmd, tant de les internes com de les externes. Per altra banda, per obtenir una informació detallada de cada comanda i dels possibles paràmetres es pot també recórrer a executar, des de la línia de comandes : *help comanda*.





A Windows es poden iniciar diferents intèrprets de comandes en el mateix sistema i mantenir-los en execució en finestres diferents de l'entorn gràfic. Aquests intèrprets funcionaran de forma independent, de manera que si es fan canvis en la configuració o en l'entorn d'un, els altres no es veuran afectats.

També és possible nidar un intèrpret de comandes dins de l'execució d'un altre, fins un màxim de 32 nivells. A primera vista això pot semblar una mica enrevessat, però permet fer canvis a la configuració i a l'entorn de la línia de comandes nidada sense afectar a la línia de comandes primària. Per anidar un intèrpret de comandes cal escriure CMD a la línia de comandes i prémer INTRO, de manera que es generarà un nou intèrpret dins de l'anterior (el qual no estarà disponible fins que no acabi l'anidat). El nou intèrpret hereta per defecte tota la configuració i l'entorn de l'intèrpret primari.

La comanda CMD també permet iniciar un nou intèrpret anidat sobre el que s'executen les comandes indicades, amb el modificador /C. Amb aquest modificador l'intèrpret anidat acaba un cop executada la comanda.

CMD /C comanda

Per acabar un intèrpret de comandes anidat s'utilitza la comanda EXIT sense arguments. Si s'introdueix aquesta comanda en un intèrpret no anidat, aquest finalitza i es tanca la finestra de la línia de comandes a l'entorn gràfic.

En el cas que es vulgui iniciar un nou intèrpret de comandes independent de forma que es mostri en una finestra diferent a la línia de comandes actual, cal utilitzar la comanda START.

3.- El registre del sistema

Els sistemes Microsoft Windows disposen d'un gran magatzem d'informació on es guarda una base de dades amb els paràmetres de configuració del sistema per a un o varis usuaris, aplicacions i dispositius. Aquest magatzem és conegut com a registre del sistema.

El registre conté informació que Windows utilitza constantment per a poder funcionar, com ara la configuració dels usuaris, les aplicacions instal·lades, els dispositius instal·lats a l'equip, etc...

El registre del sistema és una versió més completa i centralitzada dels arxius de configuració INI, AUTOEXEC.BAT i CONFIG.SYS, arxius que s'utilitzaven en les primeres versions de Microsoft Windows. Aquests arxius actualment es mantenen per compatibilitat amb aplicacions antigues que funcionaven sobre Windows, tot i que pràcticament no s'utilitzen. El registre del sistema té l'avantatge de que emmagatzema la informació de forma centralitzada i ordenada, mantenint una base de dades amb una estructura jeràrquica (en forma de carpetes o claus).

El sistema operatiu i les aplicacions que hi estan instal·lades disposen de les eines necessàries per llegir o modificar el contingut del registre del sistema. També és possible accedir directament al registre per llegir el seu contingut o modificar-lo utilitzant les comandes REGEDIT (per a sistemes operatius i aplicacions de 16 bits) o REGEDT32 (de 32 bits). La taula que es veu a continuació resumeix les claus principals que mostren els editors de registre del sistema.

Nombre	Descripción
HKEY_CURRENT_USER HKCU	Contiene la configuración del usuario que ha iniciado sesión en el equipo. Aquí se almacena información sobre las carpetas del usuario, los colores de pantalla, la configuración del <i>Panel de control</i> , etc.
HKEY_USERS HKU	Contiene la configuración de todos los usuarios definidos en el sistema. <i>HKEY_CURRENT_USER</i> es una subclave de <i>HKEY_USERS</i> .
HKEY_LOCAL_MACHINE HKLM	Contiene información sobre la configuración específica del equipo (para cualquier usuario).
HKEY_CLASSES_ROOT HKCR	Es una subclave de la subclave <i>Software</i> que a su vez pertenece a <i>HKEY_LOCAL_MACHINE</i> . La información que se almacena aquí garantiza que cuando abra una carpeta mediante el Explorador de Windows, se abrirá el programa correcto. En Windows 2000, esta información se almacena en dos claves: <i>HKEY_LOCAL_MACHINE</i> y <i>HKEY_CURRENT_USER</i> .
HKEY_CURRENT_CONFIG	Contiene información sobre la configuración de los dispositivos <i>hardware</i> que utiliza el equipo local cuando se inicia el sistema.

4.- Variables d'entorn

Les variables d'entorn en els sistemes Microsoft Windows determinen el comportament del sistema operatiu, la línia de comandes i els programes d'aplicació. També permeten obtenir informació del sistema d'una manera senzilla i utilitzar-la en comandes i programes. Existeixen tres tipus de variables d'entorn :

Del sistema

Defineixen el comportament global del sistema operatiu i els seus valors estan disponibles pels processos en execució del sistema operatiu. Només un usuari amb permisos d'administrador del sistema pot canviar els valors d'aquestes variables d'entorn.

D'usuaris

Defineixen el comportament de la línia de comandes per a l'usuari que ha iniciat sessió al sistema. Aquestes variables estan associades amb usuaris

concrets, de manera que no prenen els seus valors fins que l'usuari iniciï sessió a l'equip.

Locals

Són variables en les quals els valors s'estableixen a la línia de comandes i no són accessibles des d'altres intèrprets, a no ser que siguin intèrprets anidats.

Per poder mostrar, canviar el valor o eliminar una variable d'entorn a la línia de comandes de Microsoft Windows s'utilitza la comanda SET.

Si s'inclou algun dels caràcters <>|&^ com a valor de la variable, cal incloure el caràcter ^ davant de cadascun d'ells o bé, escriure tot el valor entre cometes dobles. Si s'utilitzen les cometes dobles, aquestes també formaran part del valor de la variable d'entorn, ja que l'intèrpret de comandes assigna a la variable tot allò que hi ha després del símbol =. No obstant, el símbol de ^ és un símbol d'escap, de forma que no apareixerà en el valor assignat a la variable d'entorn.

La comanda SET gestiona per defecte els valors assignats a les variables d'entorn com a cadenes de text, encara que siguin nombres. Per tractar els valors com a nombres cal utilitzar l'argument /A de la comanda SET. Amb aquest modificador també es poden portar a terme operacions aritmètiques senzilles, tal com es pot apreciar en les imatges següents. La primera imatge, la que veiem tot seguit, mostra els operadors numèrics de quan s'aplica la comanda SET /A :

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
%	Resto de la división entera
=	Asignación
+=	Suma y asignación
-=	Resta y asignación
*=	Multiplicación y asignación
/=	División y asignación
%=	Resto de división entera y asignación



NOTA

Cuando se trabaja con el modificador */A* del comando *SET*, todos los espacios en blanco incluidos en las asignaciones son ignorados. Esto no ocurre cuando no se utiliza este modificador.

A continuació veiem uns quants exemples :



EJEMPLO

Son operaciones numéricas del comando *SET* las siguientes:

```
SET /A Q=150
SET /A W=Q-100
SET /A Q+=5
SET /A E=Q/5
SET /A E-=1
```

Después de ejecutar todas estas asignaciones, tenemos que *Q=155*, *W=50* y *E=30*.

Es pot substituir una variable pel seu valor a la línia de comandes utilitzant el caràcter % al principi i al final del nom de la variable. Això farà que l'interpret de comandes substitueixi el nom de la variable pel seu valor abans d'analitzar la comanda i executar-la. Vegem-ne el següent exemple :



EJEMPLO

Las siguientes líneas de comandos ilustran la sustitución de las variables de entorno por sus valores. En este caso, al final se ejecuta el comando *DIR /S* para mostrar el contenido de la carpeta actual y todas sus subcarpetas:

```
C:\> SET HOLA=DIR /S
C:\> %HOLA%
```

La variable de entorno se llama "HOLA" y es sustituida en la segunda instrucción por su valor, que es "DIR /S".

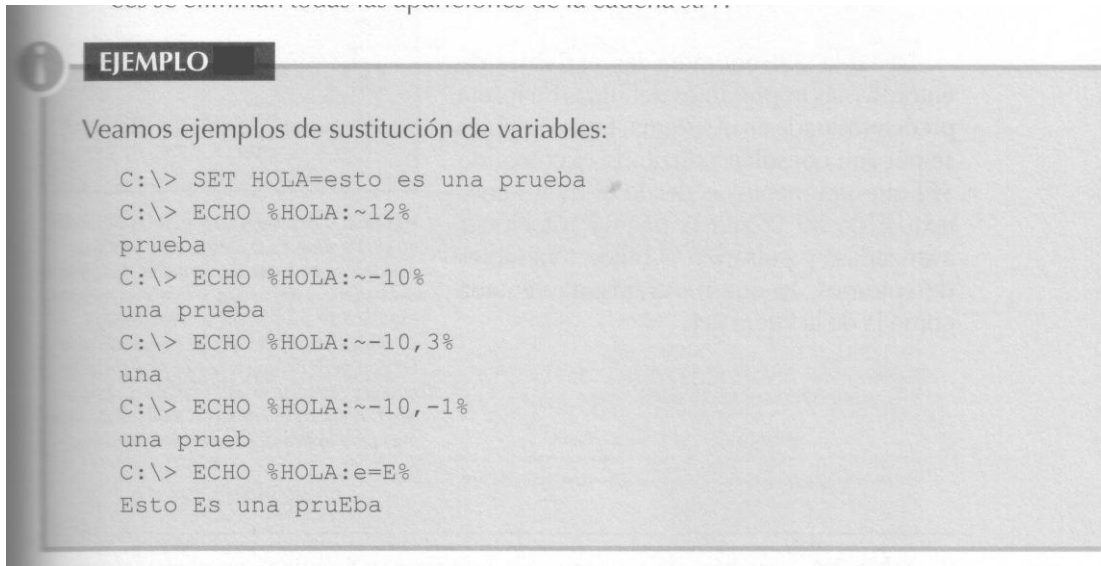
A les últimes versions de Microsoft Windows també es poden utilitzar les següents substitucions avançades de valors de variables, que es tracten com a cadenes de text :

%variable: ~n,m%

--> substitueix per la sotscadena que va des de la posició *n* (comptant des del 0) i prenent *m* caràcters del valor de la variable. Si ometem *n* o *m*, es suposa que s'agafa des del principi o fins el final del valor, respectivament. Si *n* o *m* són negatius, llavors es pren el valor menys aquest nombre.

`%variable:str1=str2%`

--> substitueix la variable pel seu valor i canvia totes les aparicions de la cadena str1 per la cadena str2. Si no s'indica str2 llavors s'eliminen totes les aparicions de la cadena str1.



En l'interpret CMD la substitució de les variables pels seus valors utilitzant el caràcter "%" es fa per defecte abans d'executar aquesta instrucció, és a dir, de forma **estàtica**. Si volem que la substitució no es faci fins que s'executi la comanda, és a dir, de forma **dinàmica**, cal utilitzar en la variable el caràcter "!" (mentre de %), i a més a més s'ha d'iniciar l'interpret amb la comanda *CMD /V:ON /C*, ja que aquest comportament no és l'establert per defecte. La substitució de variables de forma estàtica pot donar resultats totalment diferents respecte la substitució de variables en forma estàtica. Això és especialment delicat quan les variables d'entorn es troben dins d'estructures IF i FOR.

Es recomana que més endavant, quan s'hagi vist l'apartat en que s'expliquen les estructures IF i FOR es torni a mirar l'exemple que ve a continuació, associat a l'execució en forma estàtica o dinàmica. Aquest exemple ajuda a comprendre millor aquests dos conceptes, execució amb substitució estàtica de variables versus execució amb substitució dinàmica de variables.

El següent script l'hem escrit per comptar el número de vegades que apareix la paraula %2 en el fitxer %1 (recordem que per defecte els cmd scripts s'executen amb substitució estàtica de variables) :

```
@echo off
set cont=0
for /f "tokens=*" %%L in (%1) do (
    for %%P in (%%L) do (
        if /i "%%P"=="%2" set /A cont=%cont%+1
    )
)
echo A %1 surt %cont% vegades la paraula %2
```


Si executem el script :

```
$ conta.sh vallesoccidental.txt Terrassa
```

El resultat sempre serà el mateix, ó 0 ó 1. Que sigui 0 ó 1 depèn de l'últim IF que s'executi en el bucle, si retorna cert el resultat serà 1 (això passa si l'última paraula del fitxer és Terrassa) i si retorna fals el resultat serà 0.

Això és degut a que la variable `cont` es substitueix pel seu valor “estàtic” (0) cada cop que hi ha una iteració del bucle (la substitució de `%cont` només es calcula un cop, abans d'entrar per primer cop en el bucle i a cada iteració del bucle, com que `%cont` ja s'ha calculat inicialment, no es recalcula). Per tant, la sentència IF sempre s'executa de forma “estàtica” tal com podem veure a continuació :

```
if /I “%%P”==”%2” set /A cont=0+1
```

Per aquest motiu el resultat sempre és 0 ó 1 paraules trobades en el fitxer, encara que hi hagi 200 ocurrències de la paraula en el fitxer.

Per tal que `%cont` es recalculi a cada iteració és necessari utilitzar la substitució dinàmica de variables quan s'especifica la variable `cont` en la sentència IF, de manera que l'hem d'especificar de la següent manera :

```
if /I “%%P”==”%2” set /A cont=!cont!+1
```

A més a més, per tal que la substitució dinàmica de variables funcioni en el script cal establir mitjançant `SETLOCAL` el modus d'execució `ENABLEDELAYEDEXPANSION` i també s'ha d'executar el script a través d'una shell que tingui habilitada la substitució dinàmica de variables, lo qual s'habilita executant-lo invocant :

```
$ cmd /v:on /c conta.sh vallesoccidental.txt Terrassa
```

Moltes variables d'entorn estan definides de forma predeterminada en el sistema i els seus valors s'estableixen quan aquest arranca. L'ordre de prioritats que s'estableix en aplicar el valor a les variables d'entorn definides de forma predeterminada és el següent :

- (1) Variables integrades en el sistema
- (2) Variables de sistema definides a la secció `HKEY_LOCAL_MACHINE` del registre del sistema.
- (3) Variables d'usuari definides a la secció `HKEY_CURRENT_USER` del registre del sistema
- (4) Variables d'entorn definides a l'arxiu `AUTOEXEC.BAT`.

(5) Variables d'entorn definides als arxius de comandes d'inici de sessió (config.sys, arxius *.ini).

La taula que hi ha a continuació mostra les variables d'entorn més importants, definides de forma predeterminada en el sistema.

Tabla Variables de entorno más importantes definidas en el sistema

Variable	Descripción
ALLUSERSPROFILE	Ubicación del perfil de todos los usuarios.
APPDATA	Ubicación en que las aplicaciones guardan los datos de forma predeterminada.
CD	Cadena de la carpeta actual.
CMDCMDLINE	Línea de comandos exacta utilizada para iniciar el <i>CMD.EXE</i> actual.
COMPUTERNAME	Nombre del equipo.
COMSPEC	Ruta de acceso exacta al ejecutable del shell de comandos <i>CMD.EXE</i> .
DATE	Fecha actual del sistema. Utiliza el mismo formato que el comando <i>DATE /T</i> .
ERRORLEVEL	Código de error del último comando utilizado. Usualmente, los valores distintos de cero indican que se ha producido un error.
HOMEDRIVE	Letra de unidad donde se encuentra la carpeta principal del usuario.

HOMEPATH	Ruta de acceso completa de la carpeta principal del usuario.
HOMESHARE	Ruta de acceso de red de la carpeta principal compartido del usuario.
LOGONSERVER	Nombre del controlador de dominio que validó la sesión actual.
NUMBER_OF_PROCESSORS	Número de procesadores instalados en el equipo.
OS	Nombre del sistema operativo. En Windows 2000 se muestra el sistema operativo Windows NT.
PATH	Ruta de acceso de búsqueda para los archivos ejecutables.
PATHEXT	Lista de extensiones de archivo que el sistema operativo considera como ejecutables.
PROCESSOR_ARCHITECTURE	Arquitectura de chip del procesador. Los valores pueden ser: x86 o IA64 (basado en Itanium).
PROCESSOR_IDENTIFIER	Descripción del procesador.
PROCESSOR_LEVEL	Número de modelo del procesador instalado en el equipo.
PROCESSOR_REVISION	Número de revisión del procesador.
PROMPT	Configuración del símbolo del sistema del intérprete actual.
RANDOM	Número decimal aleatorio entre cero y 32767.
SYSTEMDRIVE	Unidad que contiene el directorio raíz de instalación del sistema operativo Windows.
SYSTEMROOT	Ubicación de la carpeta del sistema operativo Windows.
TEMP TMP	Carpeta temporal predeterminada que utilizan las aplicaciones disponibles. Algunas aplicaciones requieren <i>TEMP</i> y otras <i>TMP</i> .
TIME	Hora actual del sistema. Utiliza el mismo formato que el comando <i>TIME /T</i> .
USERDOMAIN	Nombre del dominio que contiene la cuenta de usuario.
USERNAME	Nombre del usuario que ha iniciado la sesión actual.
USERPROFILE	Ubicación del perfil del usuario actual.
WINDIR	Ubicación de la carpeta del sistema operativo.

Aquestes variables es poden consultar utilitzant la comanda SET sense arguments o des del menú contextual “El meu PC”, apartat “Opcions avançades” i prement el botó “Variables d'entorn”, de manera que es mostrarà (cas de Windows XP, és anàleg per altres versions de Windows) :

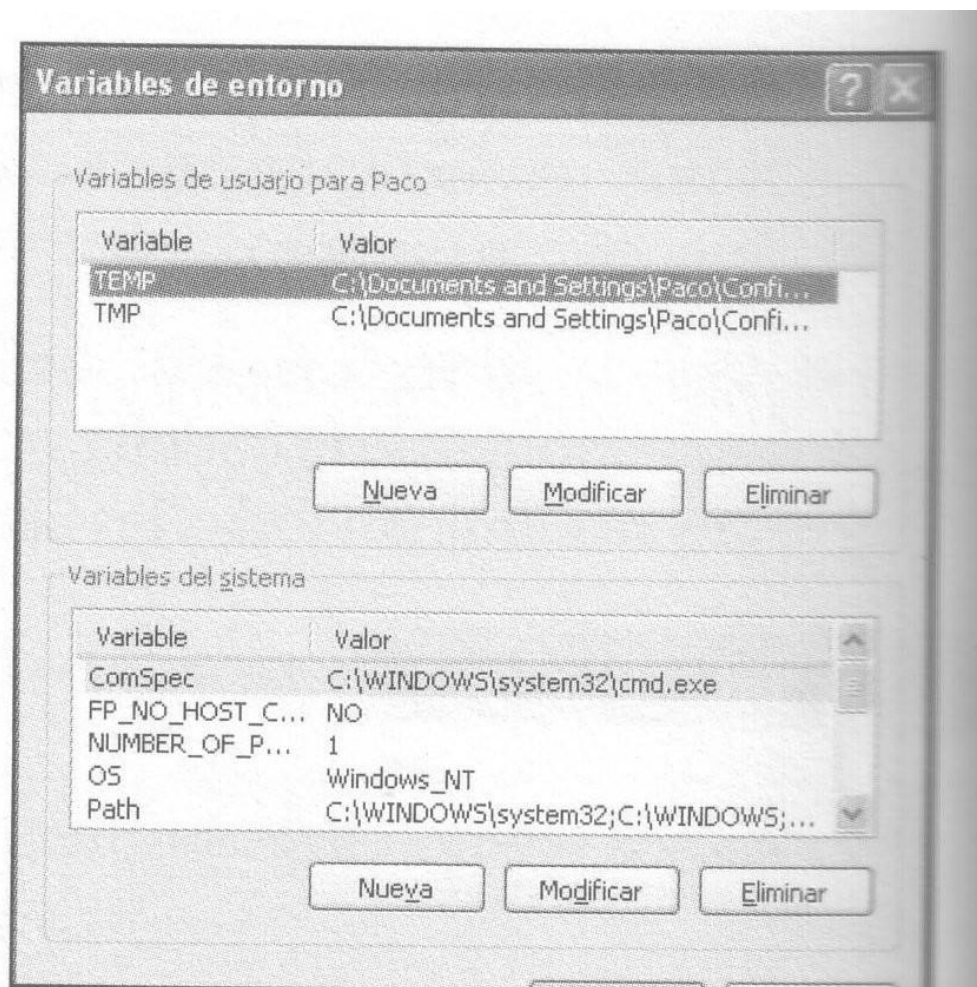


Figura Acceso a las variables de entorno de Windows XP. Desde esta ventana pueden consultarse las variables de entorno del sistema y del usuario. También pueden modificarse sus valores o crear nuevas variables, pero para ello es necesario contar con permisos para poder realizar estas tareas y reiniciar sesión (variables de usuario) o reiniciar el equipo (variables de sistema) para aplicar los cambios realizados.

Les variables *CD*, *DATE*, *TIME*, *RANDOM*, *ERRORLEVEL*, *CMDEXTVERSION* i *CMDCMDLINE* són **variables d'entorn dinàmiques**, ja que els seus valors van canviant amb el temps o amb l'estat del sistema a cada moment.

5. Redireccionaments i tuberíes

Tal com ja sabeu (repàs) la redirecció consisteix en modificar els tres tipus de fluxes d'informació amb els que treballa un programa : **les dades d'entrada**, **les dades de sortida** i els **missatges d'error**. Els programes es dissenyen utilitzant aquests tres tipus de fluxes de dades, que normalment són el teclat per a l'entrada i la pantalla per a la sortida i els errors. Quan el sistema operatiu rep una comanda, primer llegeix l'entrada, després executa la comanda i finalment (o durant l'execució) mostra la sortida i els errors.

Als fluxes d'entrada i sortida també se'ls coneix com a **control·ladors** o **dispositius**, ja que estan associats amb dispositius d'entrada i sortida connectats a l'equip. Els dispositius utilitzats per la majoria dels sistemes operatius es mostren a continuació :

Número	Nom	Descripció
0	STDIN	Entrada (per defecte el teclat)
1	STDOUT	Sortida (per defecte la pantalla o una finestra)
2	STDERR	Errors (per defecte la pantalla o una finestra)
3-9	No definits	Definits pel programa que s'executa

Els sistemes operatius moderns permeten canviar, pels programes que s'executen, aquests tres fluxes d'informació, de manera que poden ser redirigits a un altre lloc. Aquest canvi es fa utilitzant una sintaxis especial a la línia de comandes i sense necessitat de canviar el codi de les comandes que s'executaran. La re direcció dels fluxes d'informació d'una comanda pot fer-se utilitzant un arxiu o una altra comanda. L'operació de redirigir l'entrada o la sortida rep el nom de **tuberia**, **canalització**, **filtre** o **pipe**.

La forma que utilitzen la majoria de sistemes operatius amb la línia de comandes per a redirigir la informació d'entrada o sortida d'una comanda és la següent :

Redireccionament d'entrada des d'un arxiu : s'utilitza el símbol menor que (<) per a indicar que la comanda rebrà les dades d'entrada des d'un arxiu. La sintaxis és la següent:

comanda < arxiu

Redireccionament de sortida a un arxiu nou : s'utilitza el símbol major que (>) per a identificar que les dades i missatges de sortida de la comanda s'enviaran a un arxiu. Si l'arxiu indicat no existeix llavors es crea ; per altra banda, si ja existeix, llavors es perdrà el contingut anterior. La sintaxis és la següent :

comanda > arxiu

Redireccionament de sortida a un arxiu existent : s'utilitzen dos símbols major que (>>) per a indicar que les dades i missatges de sortida de la comanda s'enviaran a un arxiu. Si l'arxiu indicat no existeix, aleshores es crea ; per l'altra banda, si ja existeix, llavors les dades i missatges de sortida s'afegeixen al seu contingut. La sintaxis és la següent :

comanda >> arxiu

Gràcies als filtres, tuberíes o també anomenats *pipes*, es pot enviar la sortida d'una comanda cap a una altra comanda a la mateixa línia. Aquesta operació es pot encadenar varies vegades amb múltiples comandes, de manera que dona la sensació de que la informació es processa a través de les comandes fins que es mostra el seu resultat final.

Això és comparable a una cadena de muntatge, on el producte va passant per diferents etapes fins que finalment és muntat.

Normalment els sistemes operatius utilitzen la barra vertical | per indicar que la sortida d'una comanda (la comanda que s'escriu a l'esquerra de la barra) es redirigeixi a l'entrada de l'altra comanda (la comanda escrita a la dreta). La sintaxis genèrica és la següent :

comanda1 | comanda2 | comanda3 | ...

Anem a veure uns quants exemples :



EJEMPLO

El siguiente comando muestra el contenido del archivo "PRUEBA" pantalla a pantalla, de forma que se espera la pulsación de una tecla cuando el contenido ocupa más de una ventana:

```
C:\> MORE < PRUEBA
```

Hay que tener en cuenta que el comando *MORE* recoge cualquier cosa desde la entrada por defecto y lo muestra en varias páginas. El comando anterior es igual que este otro:

```
C:\> MORE 0< PRUEBA
```

El siguiente comando guarda en el archivo "LISTA" un listado de los archivos y carpetas de la carpeta actual:

```
C:\> DIR > LISTA
```

El comando anterior es equivalente a este otro:

```
C:\> DIR 1> LISTA
```

El siguiente comando añade el listado con el contenido de la carpeta padre de la actual al listado de los archivos y carpetas de la carpeta actual que ha sido creado anteriormente:

```
C:\> DIR .. >> LISTA
```

El comando anterior es equivalente a este otro:

```
C:\> DIR .. 1>> LISTA
```



EJEMPLO

El siguiente comando muestra el contenido de la carpeta actual y todas sus subcarpetas, y espera a que se pulse una tecla cada vez que aparece una página:

```
C:\> DIR /S | MORE
```

También se pueden crear varias tuberías o *pipes*, como en el siguiente comando, que muestra un listado ordenado del contenido de la carpeta actual en diferentes páginas:

```
C:\> DIR | SORT | MORE
```

El comando anterior se completa de esta forma: primero se ejecuta *DIR* y su salida se pasa a *SORT*, que la ordena y la pasa a *MORE*, para que finalmente la divida en diferentes páginas y la muestre por pantalla. El comando *SORT* ordena por líneas teniendo en cuenta el comienzo de cada una.



EJEMPLO

Los siguientes comandos muestran el contenido del archivo "LISTA" por pantalla de forma ordenada:

```
C:\> MORE < LISTA | SORT
```

El mismo efecto de los comandos anteriores puede ser redirigido a otro archivo llamado "LISTAORDENADA":

```
C:\> MORE < LISTA | SORT > LISTAORDENADA
```



EJEMPLO

Los siguientes comandos redirigen la salida del comando *DIR /S* al archivo "LISTA", mientras que los errores se redirigen al archivo "ERRORES":

```
C:\> DIR /S > LISTA 2> ERRORES
```

Los siguientes comandos guardan en el archivo "LISTA" el listado ordenado de la carpeta actual:

```
C:\> DIR | SORT > LISTA
```

Nótese que si se modifica la posición de los comandos, el efecto no es el mismo, ya que la ejecución de éstos se realiza en otro orden:

```
C:\> DIR > LISTA | SORT
```

A la línea de comandos de Microsoft Windows también es posible redirigir un dispositivo d'una comanda a un altre dispositiu. Per a això s'utilitza la següent sintaxis :

`n1>&n2`

En l'expressió anterior s'ha de substituir `n1` pel control·lador que envia la sortida i `n2` pel controlador que la rep. Per exemple, lo següent envia els errors d'execució de la comanda (`STDERR`) a la sortida per defecte (`STDOUT`) :

```
C:\> DIR 2>&1
```

Aquesta característica per si sola no és molt útil, ja que a l'exemple anterior, en executar la comanda *DIR*, per defecte ja s'envien els errors a la pantalla. No obstant, sí que resulta molt efectiva quan volem duplicar un dispositiu.

A la línia de comandos és possible duplicar un dispositiu, tant per l'entrada com per la sortida. Si el dispositiu es duplica per a l'entrada, llavors pot acceptar dades d'entrada des de varies fonts. Per altra banda, si el dispositiu es duplica per a la sortida, llavors pot ser enviat a altres destins. Per duplicar un dispositiu cal especificar-lo varies vegades en la mateixa línia, tal com es mostra a l'exemple que es veu a continuació.



EJEMPLO

El siguiente comando envía su salida al archivo "LISTA" y también la envía al dispositivo de errores, por lo que también aparecerá en pantalla:

```
C:\> DIR 1> LISTA 1>&2
```

Por su parte, el siguiente comando envía su salida y sus errores al mismo archivo "RESULTADO":

```
C:\> DIR 1> RESULTADO 2>&1
```

En el comando anterior, se ha indicado que la salida se envía al archivo "RESULTADO" (con "1>"), pero no es necesario indicar el número de dispositivo porque es el de salida por defecto. Hay que indicar la duplicación de la entrada de esta forma ya que el siguiente comando es incorrecto (se puede duplicar un dispositivo pero no un archivo):

```
C:\> DIR 1> RESULTADO 2> RESULTADO
```



NOTA

En la línea de comandos se utiliza el carácter ^ para indicar que el símbolo que va a continuación debe interpretarse como texto. Por ejemplo, si se desea asignar el texto "Hola>" a una variable de entorno, habría que escribir `SET var=Hola^>`, lo que significa que el símbolo > no debe interpretarse como una redirección, sino simplemente como un carácter de texto.

6.- Execució seqüencial i condicional de comandes

L'interpret de comandes dels sistemes Microsoft Windows no és capaç d'executar vàries comandes de forma simultània, tot i que es pot aconseguir si s'executen en diferents intèrprets independents, ja que el sistema operatiu sí que és multitasca.

No obstant, la línia de comandes de Windows permet l'execució seqüencial i condicional de comandes. La taula que es veu a continuació mostra la sintaxis que s'utilitza per a l'execució seqüencial i condicional de comandes.

Tabla Ejecución secuencial y condicional de comandos en Windows

Operación	Descripción
<i>comando1 & comando2</i>	Ejecuta el primer comando y cuando finaliza ejecuta el segundo comando.
<i>comando1 && comando2</i>	Ejecuta el primer comando y, si finaliza correctamente (variable <i>ERRORLEVEL</i> es 0), ejecuta el segundo comando.
<i>comando1 comando2</i>	Ejecuta el primer comando y, si no finaliza correctamente (variable <i>ERRORLEVEL</i> es distinta de 0), ejecuta el segundo comando.