

SISTEMAS INFORMÁTICOS

PROGRAMACIÓN BATCH

INTRODUCCIÓN A LA PROGRAMACIÓN BATCH

Imagina que tienes una lista de tareas que haces en tu computadora, como abrir programas, copiar archivos, crear carpetas... ¡Aburrido hacerlas una por una, verdad?

La programación batch te permite decirle a la computadora: "Oye, haz todas estas cosas por mí, en este orden".

¿Cómo funciona?

Escribes las instrucciones en un archivo de texto, cada línea es una instrucción, como si le dieras órdenes a la computadora. Este archivo se guarda con la extensión **.bat**.

La computadora lee y ejecuta las instrucciones, cuando haces doble clic o llamas al archivo **.bat** desde la línea de comandos del **cmd**, la computadora lee las instrucciones una por una y las ejecuta en orden. Un ejemplo de fichero bat, sería el siguiente:

```
@echo off  
echo SE VAN A INICIAR LAS APLICACIONES CHROME Y NOTEPAD  
start chrome  
start notepad
```

INTRODUCCIÓN A LA PROGRAMACIÓN BATCH

Comandos: Son las palabras clave que le dicen a la computadora qué hacer (ejemplos: `start`, `copy`, `mkdir`, `echo`).

Parámetros: Información adicional que se le da a un comando (ejemplo: en `start chrome`, "chrome" es el parámetro que indica qué programa abrir).

Variables: Son como contenedores donde guardar información (como el nombre de un archivo, una ruta, una IP o un valor introducido por teclado).

Lógica condicional: Permite que el script tome decisiones (ejemplos: `if`, `else`).

Bucles: Permiten repetir un conjunto de instrucciones varias veces (ejemplo: `for`).

¿Para qué sirven estos elementos?

- **Automatizar tareas repetitivas:** Ahorra tiempo y esfuerzo.
- **Simplificar tareas complejas:** Combinar varios comandos en un solo script.
- **Crear herramientas personalizadas:** Adaptar el comportamiento de la computadora a tus necesidades.

COMANDOS FICHEROS BAT

echo: Muestra texto en la pantalla.

echo Hola mundo - Muestra el mensaje "Hola mundo" en la consola.

echo . - Imprime una línea en blanco, útil para separar visualmente la información.

@echo off - Desactiva la visualización de los comandos en la consola, haciendo que solo se vea la salida del script. Se suele colocar al principio del archivo .bat.

pause: Detiene la ejecución del script y espera a que el usuario presione una tecla.

pause - Muestra el mensaje "Presione una tecla para continuar..." y pausa la ejecución hasta que se presione una tecla. Útil para que el usuario pueda leer la salida del script antes de que se cierre la ventana.

cls: Limpia la pantalla de la consola.

COMANDOS FICHEROS BAT

rem o **::**: Permite agregar comentarios al código. Los comentarios son ignorados por el intérprete de comandos y sirven para explicar el código o dejar notas para el programador.

`rem Este es un comentario` - El texto "Este es un comentario" no se ejecutará.

`:: Otro comentario` - Igual que `rem`, se utiliza para agregar comentarios.

title: Cambia el título de la ventana de la consola.

`title Mi Script` - Cambia el título de la ventana a "Mi Script".

color: Cambia el color del texto y del fondo de la consola.

`color 0a` - Cambia el color del texto a verde (a) sobre fondo negro (0). Puedes consultar la lista de colores disponibles escribiendo `color /?` en la consola.

COMANDOS FICHEROS BAT

start: Abre un programa o archivo.

start notepad - Abre el Bloc de notas.

start "Mi documento" "C:\Documentos\mi_documento.txt" - Abre el archivo "mi_documento.txt" con el título "Mi documento" en la ventana.

dir: Muestra una lista de archivos y carpetas en el directorio actual.

dir - Muestra el contenido del directorio actual.

dir /s - Muestra el contenido del directorio actual y de todos sus subdirectorios.

cd: Cambia el directorio actual.

cd C:\Documentos - Cambia el directorio actual a "C:\Documentos".

cd .. - Sube un nivel en la estructura de directorios.

COMANDOS FICHEROS BAT

mkdir: Crea un nuevo directorio.

`mkdir Nueva_Carpeta` - Crea una carpeta llamada "Nueva_Carpeta" en el directorio actual.

copy: Copia archivos.

`copy archivo.txt C:\Nueva_Carpeta` - Copia el archivo "archivo.txt" a la carpeta "C:\Nueva_Carpeta".

move: Mueve o renombra archivos o directorios.

`move archivo.txt C:\Nueva_Carpeta` - Mueve el archivo "archivo.txt" a la carpeta "C:\Nueva_Carpeta".

`move archivo_viejo.txt archivo_nuevo.txt` - Renombra el archivo "archivo_viejo.txt" a "archivo_nuevo.txt".

INTRODUCCIÓN A LA PROGRAMACIÓN BATCH

del: Elimina archivos.

`del archivo.txt` - Elimina el archivo "archivo.txt".

`del *.tmp` - Elimina todos los archivos con la extensión ".tmp" en el directorio actual.

rmdir: Elimina un directorio.

`rmdir Nueva_Carpeta` - Elimina la carpeta "Nueva_Carpeta" (si está vacía).

`rmdir /s /q Nueva_Carpeta` - Elimina la carpeta "Nueva_Carpeta" y su contenido sin preguntar.

Estos son solo algunos de los comandos más utilizados en la programación batch. Hay muchos otros comandos disponibles, y puedes combinarlos para crear scripts complejos que automaticen tareas y realicen operaciones avanzadas.

CONTROL DE FLUJO (Bifurcaciones e Iteraciones)

if: Permite ejecutar comandos condicionalmente.

`if exist archivo.txt echo El archivo existe` - Muestra el mensaje "El archivo existe" si el archivo "archivo.txt" existe.

```
@echo off
set /p nombre="Introduce tu nombre: "
if "%nombre%"=="Juan" ( echo Hola Juan! ) else ( echo Hola %nombre%! )
```

```
@echo off
if defined mi_variable ( echo La variable mi_variable está definida. )
else ( echo La variable mi_variable no está definida. )
```

```
@echo off
set /p numero="Introduce un número: "
if %numero% gtr 10 ( echo El número es mayor que 10. ) else ( echo El número es menor o igual que 10. )
```

INTRODUCCIÓN A LA PROGRAMACIÓN BATCH

for: Permite repetir un conjunto de comandos para cada elemento de una lista.

`for %a in (*.txt) do echo %a` - Muestra el nombre de cada archivo con extensión ".txt" en el directorio actual.

goto: Posibilita controlar el salto de ejecución del script, a una etiqueta que hayamos colocado en otra parte del código.

`goto :menu` - La secuencia de ejecución del script, al pasar por esta línea será direccionada a la etiqueta menu.

PARÁMETROS

Provenientes del exterior: Son valores que se pasan al script batch cuando se ejecuta. Se escriben después del nombre del archivo .bat en la línea de comandos.

```
mi_script.bat parametro1 parametro2
```

Acceso mediante %1, %2, etc.: Se accede a los parámetros dentro del script utilizando %1 para el primer parámetro, %2 para el segundo, y así sucesivamente.

Predefinidos: Algunos parámetros especiales como %0 (nombre del script), %date% (fecha actual), %time% (hora actual) ya están definidos.

De solo lectura: No se pueden modificar los valores de los parámetros dentro del script.

```
@echo off
echo El nombre del script es: %0
echo El primer parámetro es: %1
set mi_variable=Hola
echo %mi_variable% mundo!
```

VARIABLES

Definidas dentro del script: Se crean y asignan valores dentro del script utilizando el comando **set**.

```
set mi_variable=valor
```

Acceso mediante %nombre_variable%: Se accede a las variables utilizando el nombre de la variable entre signos de porcentaje.

```
echo %mi_variable%
```

Modificables: Se pueden cambiar los valores de las variables dentro del script.

Alcance local: Por defecto, las variables solo existen dentro del script donde se definen.

COMANDOS POR TIPO

Información del sistema:

- `systeminfo` : Muestra información detallada sobre la configuración del sistema.
- `ver` : Muestra la versión de Windows.
- `ipconfig` : Muestra la configuración de red del equipo.
 - Ejemplo: `ipconfig /all` (muestra toda la información de la configuración de red)
- `tasklist` : Muestra una lista de los procesos en ejecución.
- `driverquery` : Lista los controladores instalados.

Administración del sistema:

- `shutdown` : Apaga o reinicia el equipo.
 - Ejemplo: `shutdown /r /t 0` (reinicia el equipo inmediatamente)
- `sfc /scannow` : Escanea y repara archivos de sistema corruptos.
- `chkdsk` : Comprueba la integridad del disco duro y corrige errores.
 - Ejemplo: `chkdsk C: /f /r` (comprueba la unidad C:, corrige errores y recupera sectores dañados)
- `gpupdate` : Actualiza la configuración de la directiva de grupo.

Redes:

- `ping` : Envía paquetes ICMP a un host para comprobar la conectividad.
 - Ejemplo: `ping www.google.com`
- `tracert` : Muestra la ruta que toman los paquetes para llegar a un host.
 - Ejemplo: `tracert www.google.com`
- `netstat` : Muestra las conexiones de red activas.

Otros comandos útiles:

- `cls` : Limpia la pantalla de la consola.
- `exit` : Cierra la ventana de CMD.
- `help` : Muestra ayuda sobre un comando específico.
 - Ejemplo: `help dir`
- `date` : Muestra y permite cambiar la fecha del sistema.
- `time` : Muestra y permite cambiar la hora del sistema.

MENÚ OPCIONES

```
@echo off
cls
:menu
cls
echo.
echo 1. Abrir Google Chrome maximizado
echo 2. Abrir Administrador de equipos
echo 3. Hacer ping a Google
echo 4. Salir echo.

choice c/ 1234 /M "Elige una opción:"

if errorlevel 4 exit
if errorlevel 3 goto :ping
if errorlevel 2 goto :admin
if errorlevel 1 goto :chrome

:chrome
    start chrome "https://www.google.com/" --start-maximized
    goto :menu
:admin
    start compmgmt.msc
    goto :menu

:ping
    ping www.google.com
    goto :menu
```

FILTRO DE ARCHIVOS POR CARPETAS

```
@echo off  
cls
```

```
REM Directorio base para la estructura de carpetas  
set "directorio_base=C:\Nueva_Estructura"
```

```
REM Crear las carpetas  
mkdir "%directorio_base%"  
mkdir "%directorio_base%\Documentos"  
mkdir "%directorio_base%\Imágenes"  
mkdir "%directorio_base%\Videos"
```

```
REM Copiar los archivos a sus respectivas carpetas  
for %%a in (*.txt *.doc *.docx *.pdf) do move "%%a" "%directorio_base%\Documentos"  
for %%a in (*.jpg *.jpeg *.png *.gif) do move "%%a" "%directorio_base%\Imágenes"  
for %%a in (*.mp4 *.avi *.mkv) do move "%%a" "%directorio_base%\Videos"
```

```
echo Estructura de carpetas creada y archivos copiados correctamente.  
pause
```


DATOS ARCHIVOS Y VARIABLES ENTORNO

```
@echo off
rem Modificadores de un argumento
echo Ruta al archivo: %~f1
echo Disco: %~d1
echo Solo ruta: %~p1
echo Nombre: %~n1
echo Extension: %~x1
echo Ruta Corta: %~s1
echo Atributos: %~a1
echo Fecha: %~t1
echo Tamaño: %~z1
```

%PROGRAMFILES% : Ruta al directorio de archivos de programa (generalmente C:\Program Files).

%PROGRAMFILES(X86)% : Ruta al directorio de archivos de programa para aplicaciones de 32 bits en sistemas de 64 bits.

%PROMPT% : Cadena que define el prompt de la consola.

%SYSTEMDRIVE% : Letra de la unidad donde está instalado Windows (generalmente C:).

%SYSTEMROOT% : Ruta al directorio de Windows (generalmente C:\Windows).

%TEMP% : Ruta al directorio temporal del usuario actual.

%TIME% : Hora actual en formato hh:mm:ss.nn.

%USERNAME% : Nombre de usuario actual.

%USERPROFILE% : Ruta al directorio de perfil del usuario actual.

%WINDIR% : Ruta al directorio de Windows (igual que **%SYSTEMROOT%**).

%ALLUSERSPROFILE% : Ruta al directorio de perfil de todos los usuarios.

%APPDATA% : Ruta al directorio de datos de aplicación del usuario actual.

%COMPUTERNAME% : Nombre del equipo.

%COMSPEC% : Ruta al intérprete de comandos (cmd.exe).

%DATE% : Fecha actual en formato dd/mm/aaaa.

%HOMEDRIVE% : Letra de la unidad donde se encuentra el perfil del usuario (generalmente C:).

%HOMEPATH% : Ruta al directorio principal del usuario.

%NUMBER_OF_PROCESSORS% : Número de procesadores del sistema.

%OS% : Nombre del sistema operativo (ej. Windows_NT).

%PATH% : Lista de directorios donde se buscan los archivos ejecutables.

%PATHEXT% : Extensiones de archivo que se consideran ejecutables.

%PROCESSOR_ARCHITECTURE% : Arquitectura del procesador (ej. x86 o AMD64).

%PROCESSOR_IDENTIFIER% : Identificador del procesador.

%PROCESSOR_LEVEL% : Nivel del procesador.

%PROCESSOR_REVISION% : Revisión del procesador.

%PROGRAMDATA% : Ruta al directorio de datos de programa compartidos por todos los usuarios.

**Gracias por tu
atención**