

# Unidad 3

## Batch Scripting



# Lenguaje batch tutorial y comandos básicos .bat

Este es un tutorial sencillo para principiantes en el que se intenta explicar de la forma más simple los principales comandos.

El lenguaje batch no es muy difícil pero para personas que se estén iniciando puede resultar un poco complicado así que intentare explicarlo de la forma más sencilla posible.

¿Qué es batch?

Un archivo batch es un archivo de texto sin formato guardado con la extensión bat y que contiene un conjunto de ordenes llamadas comandos de DOS.

En resumen un archivo batch es un archivo de texto en el que se ha escrito un conjunto de códigos que van a ser ejecutados de forma lineal (ya lo irán entendiendo con ejemplos más adelante) y ese archivo de texto debe ser guardado con la extensión .bat, es decir un nombre cualquiera.bat ej: nombredelarchivo.bat

Es muy útil para automatizar tareas, por ejemplo un programa que nos abra todas las herramientas que necesitamos para trabajar en un determinado proyecto, de esta forma cada vez que queramos trabajar solo tendremos que pulsar un botón y no hará falta abrir una a una todas las aplicaciones necesarias, pero su utilidad no se queda en esto, todo hacker debe conocerlo, especialmente su equivalente en linux: los scripts de bash.



La terminación bat viene del propio nombre del lenguaje (batch) que debido a que las extensiones de archivos (.bat, .exe, .txt etc) solo pueden tener 3 letras quedo reducido a eso.

## PARTE 1

Lo primero que debemos hacer es abrir un archivo de texto y guardarlo con un nombre seguido de .bat tal como expliqué anteriormente en ese archivo vamos a poner nuestros comandos.

Empecemos quitando el eco, para eso escribimos: @echo off

```
@echo off  
pause  
exit
```

Si no sabes lo que es el eco prueba a quitar @echo off y veras lo que pasa.

Pause sirve para hacer una pausa en la secuencia (pulsa una tecla para continuar)  
y exit sirve para finalizar



Bien ahora vamos a continuar con otro código: el echo y el echo.  
(nótese el punto)

El echo sirve para escribir algo en pantalla por ejemplo echo hola  
escribiría hola en la pantalla  
y echo. sirve para dejar un espacio

```
@echo off  
echo.  
echo Hola Instituto #root  
echo.  
echo Como estas?  
pause  
exit
```

Prueba ponerlo y comprueba lo que pasa

Es muy importante que vayas entendiendo todo esto paso a paso  
sino se irá complicando bastante.

Ahora vamos a ver una variación del pause: pause>NUL, sirve para  
hacer una pausa sin poner nada en la pantalla, puedes probarlo  
sustituyendo pause por pause>NUL en los códigos anteriores



Esto lo vamos a utilizar para poner nosotros el mensaje que queramos en vez del “pulsa una tecla para continuar”  
y lo vamos a hacer poniendo algo seguido de pause>NUL veréis el efecto

```
@echo off
```

```
echo.
```

```
echo Pulsa una tecla para ir a la siguiente parte del tutorial de  
Instituto #root
```

```
pause>nul
```

```
echo.
```

```
echo Bienvenido a la segunda parte del tutorial
```

```
echo.
```

```
echo Pulsa una tecla para salir
```

```
pause>nul
```

```
exit
```

## PARTE 2

El comando cls sirve para borrar la pantalla.





```
@echo off
echo.
echo Pulsa una tecla para ir a la siguiente parte del tutorial
pause>nul
cls
echo.
echo Bienvenido a la segunda parte del tutorial de Instituto
#root
echo.
echo Pulsa una tecla para salir
pause>nul
exit
```

Este es muy sencillo, ejecútalo y te darás cuenta de su utilidad.

Ahora vamos a pasar a las variables, algo que tienen casi todos los lenguajes de programación y que, como verás, te da infinitas posibilidades.

Una variable es un nombre que designa un valor y como su nombre indica puede variar. Gracias a esto podemos guardar valores y darles nombre.

Para poner una variable debemos escribir se nombrevariable=valor. ej: set nombre= pepito



y para referirnos a esa variable debemos escribir el nombre de la variable entre %% ejemplo: hola %nombre%

```
@echo off  
set nombre= Pepito  
echo.  
echo Hola %nombre%  
echo.  
pause  
exit
```

Es muy fácil pero es importante que lo entiendas, porque en esto se basa todo lo que viene a continuación, si te estas preguntando por que es realmente importante, ahora vas a verlo.

Ahora vamos a hacer que podamos poner nosotros mismos la variable, es decir que la persona que usa el programa puede asignar un valor. Para eso en vez de poner set debemos poner set/p



ej: set/p nombrevariable= valor

@echo off

echo.

set/p nombre= ¿Como te llamas?

cls

echo.

set/p años= ¿Cuantos años tienes?

cls

echo.

echo Hola %nombre% tienes %años% años.

echo.

echo Pulsa una tecla para salir.

pause>nul

exit

Una vez entendido esto vamos a empezar a utilizar las variables por ejemplo para hacer operaciones con ellas.

Para eso utilizaremos set/a. es como el set/p pero no se muestra en pantalla y realiza operaciones, porcentajes, sumas, restas etc.

Estos son los signos que podemos utilizar:

+ para sumar

– para restar

\* para multiplicar

/ para dividir





Y lo vamos a expresar así:

```
@echo off
```

```
echo.
```

```
set/p numero1= Dime un numero
```

```
cls
```

```
echo.
```

```
set/p numero2= Dime otro para sumar al anterior
```

```
cls
```

```
set/a suma= %numero1% + %numero2%
```

```
echo.
```

```
echo %numero1% + %numero2% = %suma%
```

```
echo.
```

```
echo Pulsa una tecla para salir.
```

```
pause>nul
```

```
exit
```

## PARTE 3

En esta parte del tutorial vamos a aprender a trabajar con if, if not, else y goto.

If sirve para expresar una condición. Es decir si pasa esto entonces haz esto. Para ello vamos a utilizar la siguiente estructura: if condición acción

ejemplo: if %variable%==4 echo bien!! esto quiere decir que si el valor de la variable es igual a cuatro entonces escribe “bien!!”



Las distintas condiciones que podemos comprobar con un if en batch son las siguientes: (operadores)

if variable==valor (para expresar si la variable es igual a valor haz esto)

if variable LSS valor (para expresar si la variable es menor a valor haz esto)

if variable GTR valor (para expresar si la variable es mayor a valor haz esto)

if variable LEQ valor (para expresar si la variable es menor o igual haz esto)

if variable GEQ valor (para expresar si la variable es mayor o igual haz esto)

if not se utiliza igual que if pero expresa todo lo contrario ej:

if not variable==valor (para expresar si la variable NO es igual a "valor") y lo mismo con el resto de condiciones vistas arriba.

```
@echo off
```

```
echo.
```

```
set/p variable= ¿Cuanto son 2 + 2?
```

```
cls
```

```
echo.
```

```
if %variable%==4 echo bien!!
```

```
echo.
```

```
pause
```

```
exit
```



Aquí también se podría utilizar el goto que sirve para enviar la lectura de comandos a otro sitio del bat por ejemplo if variable==valor goto correcto, después mas abajo pondríamos una “etiqueta” con ese nombre después de dos puntos ej: :correcto así quedaría: si variable es igual a valor entonces ir al menú “correcto” y en el ya escribiríamos lo que queramos con un ejemplo lo vas a entender mucho mejor.

```
@echo off
echo.
set/p variable= ¿Cuanto son 2 + 2?
cls
echo.
if %variable%==4 goto correcto
echo.
:correcto
echo bien!!
pause
exit
```

En este caso no tendría mucho sentido pero cuando tenemos más variables y muchas opciones a elegir es esencial útil ya lo iréis viendo

Este ejemplo indicaría que si la respuesta a la pregunta “¿cuánto son 2+2?” es 4 entonces escribe bien!!



Pero ¿qué pasa si la respuesta no es esa? Pues para eso sirve else, se utiliza así:

if variable==valor goto correcto else goto incorrecto esto significaría que si la respuesta es 4 entonces escribe bien pero si la respuesta es cualquier otra cosa escribe mal!!!

```
@echo off
```

```
:inicio
```

```
echo.
```

```
set/p pregunta= ¿cuanto son 2 + 2?
```

```
cls
```

```
echo.
```

```
if %pregunta%==4 goto correcto else goto incorrecto
```

```
:incorrecto
```

```
echo.
```

```
echo Mal!!
```

```
echo.
```

```
echo Vuelve a intentarlo
```

```
pause
```

```
cls
```

```
goto inicio
```

```
:correcto
```

```
echo.
```

```
echo Bien!!
```

```
echo Pulsa una tecla para salir de este ejemplo de
```

```
www.institutoroot.com
```

```
pause>nul
```

```
exit
```

