# PRÁCTICA 23

**OBJETIVO: *Utilizar condiciones dobles y en cascada.***

DESCRIPCIÓN

En todos los lenguajes de programación existen estructuras condicionadas:

* Simples
* Condición Doble.
* Condición en casca

Puede existir una condición múltiple. En fichero por lotes no existen condiciones múltiples ( SWITCH, WATCH…)

Las condiciones en CMD con ficheros por lotes avanzados, se utiliza: IF

## Modo 1: Control de errores.

IF ERRORLEVEL numero GOTO :etiqueta

## Modo 2: Comparar el valor de dos variables.

IF %vari% == %vari2% GOTO :etiqueta

Se suele acompañar de un carácter a ambos lados de la comparación para evitar que sucedan errores de comparación si una variable no tiene valor

IF %vari%caracter == %vari2%caracter GOTO :etiqueta

Ej.:

IF %vari%**.** == %vari2%. GOTO :etiqueta IF %vari%**a** == %vari2%**a** GOTO :etiqueta

## MODO 3: Se utiliza para comprobar la existencia de un archivo.

IF EXIT archivo GOTO :etiqueta

## Estos tres modos son los básicos y heredados desde las primeras versiones utilizados en ficheros BATH. MODO 4: La negación de los MODOS 1, 2, 3

IF NOT ERRORLEVEL COMANDO IF NOT %vari% == %vari2% COMANDO

IF NOT EXIT fichero COMANDO

**NOTA:** cuando especificamos COMANDO, puede ser cualquier comando, lo suyo es controlar si es un bloque independiente, en una etiqueta como si fuera una rutina o proceso. Con el único inconveniente que una vez ejecutado no retorno a la línea siguiente, ya que se utiliza GOTO :ETIQUETA.

**MODO 5: A partir de Windows XP se permite realizar en un IF una conjunto de órdenes**, siempre que se encuentren dentro de un paréntesis.

IF condición (

COMANDO 1

COMANDO 2

COMANDO 3

)

* El requisito es que el paréntesis se especifique su apertura en la condición IF (
* Los comandos se especifiquen en la siguiente línea al (
* El cierre del bloque o paréntesis, vaya en una línea al final y en blanco, solo se escriba )

## MODO 6: En Windows 7 y posteriores se pueden utilizar condiciones dobles. Si …. SINO …

IF condición (

COMANDO 1

COMANDO 2

COMANDO 3

) ELSE (

COMANDO 4

COMANDO 5

COMANDO 6

COMANDO 7

)

Esta es la sintaxis a utilizar, pero además hay que tener en cuenta que dentro de un paréntesis se puede ejecutar UN COMANDO o varios COMANDOS.

## MODO 7: CONDICIONES EN CASCADA

Ejemplo 1:

IF condición (

COMANDO 1

COMANDO 2

COMANDO 3

) ELSE (

COMANDO 4

IF condición (

COMANDO 10

COMANDO 11

) ELSE (

)

COMANDO 12

COMANDO 5

COMANDO 6

COMANDO 7

)

Ejemplo 2:

IF condición (

COMANDO 1

) ELSE (

IF condición (

COMANDO 10

COMANDO 11

) ELSE (

)

)

COMANDO 12

Ejemplo 3:

IF condición (

COMANDO 1

) ELSE (

IF condición (

COMANDO 10

COMANDO 11

) ELSE (

IF condición (

COMANDO 12

)

)

)

IF en los CMD AVANZADOS

Si los comandos de extensión están habilitados y corresponde a las tres sintaxis siguientes:

1. **MODO 8:** Condiciones utilizando las operaciones de comparación. Siempre se tiene que utilizar la opción /I seguida de IF

IF [/I] cadena1 op-de-comparación cadena2 comando donde op-de-comparación puede ser:

|  |  |  |
| --- | --- | --- |
| EQU | igual | = |
| NEQ | no igual | >< >< |
| LSS | menor que | < |
| LEQ | menor que o igual | <= |
| GTR | mayor que | > |
| GEQ | mayor que o igual | >= |

cadena Puede ser una variable de ambiente %variable%, un parámetro reemplazable %1,%2… , un valor numérico o una cadena

1. **MODO 9:** Comprobar el número de la versión IF CMDEXTVERSION número comando
2. **MODO 10:** Comprobar si una variable está definida o no. Si la variable tiene valor está definida ( es lo que se quiere comprobar con esta condición)

IF DEFINED variable comando

## PASO 1: Crear menús utilizando la comparación ERRORLEVEL como variable en una comparación avanzada

ERRORLEVEL se puede utilizar como variable de entorno para obtener o visualizar el valor del código de error que nos da cualquier comando, en este ejemplo es el comando CHOICE.

Se emplea nombre%ERRORLEVEL% GOTO :nombre%ERRORLEVEL%

La condición necesaria es que todas las etiquetas tengan la misma raíz, pero solo cambie el número, en función de las opciones de CHOICE /C abcs . Cada letra indica un número de ERRORLEVEL, si se elige la a=1, b=2, c=3, s=4

Cada opción seleccionada se asigna a %ERRORLEVEL% el valor (1,2,3,4), que se sustituye en la variable de ambiente dentro del GOTO :nombre1,..

@ECHO OFF

:REPETIR CLS ECHO.

ECHO Menu que controla las siguientes opciones ECHO ==================================== ECHO.

ECHO a. Ver la versión del sistema Operativo ECHO.

ECHO b. Ver el volumen de la Unidad C:

ECHO.

ECHO c. Visualizar las variables de ambiente ECHO.

ECHO s. Salir del programa ECHO.

CHOICE /C abcs /CS /T 30 /D s /M "Elija una opcion de las siguientes" GOTO :opcion%ERRORLEVEL%

REM

REM este goto no tiene significado en principio, salvo que ocurra un error.

REM salta al final

REM

GOTO :EOF

**:opcion3**

::vervariables

CLS

SET

PAUSE

GOTO :REPETIR

**:opcion2**

::vervolumen

CLS

VOL

PAUSE

GOTO :REPETIR

**:opcion1**

::verversion

CLS

VER

PAUSE

GOTO :REPETIR

**:opcion4**

REM SALIR ó :EOF

**PASO 2: Seleccionar, crear, ejecutar y explicar**

Acceder a la dirección : <https://gist.github.com/HiroNakamura/4702728>

a) Buscar 20 ficheros que contengan IF y clasificar según la explicación, desde el MODO 1 al 10

b) Copiar el contenido de los ficheros guardándolos con el nombre IF001.CMD, … , IF020.CMD

c) Crear un documento que explique brevemente el funcionamiento de cada IF001.CMD

d) Subir con el nombre PRACTICA 18-NombreAlumno.rar ó .zip.

NOTA: no capturar ventanas, ni gráficos, solo capturar en formato texto