

C.F.G.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA

MÓDULO:

Sistemas Informáticos

Unidad 5

**ARCHIVOS DE PROCESAMIENTOS
POR LOTES**

1. INTRODUCCIÓN.

Los Archivos por lotes son los elementos más apreciados por los usuarios de computadores ya que permiten agilizar y simplificar el trabajo diario.

En todo PC se pueden ejecutar tres tipos distintos de programas, diferenciados solamente por sus extensiones, estas extensiones de programas son: ".COM", ".EXE" y ".BAT" y recordemos que para ejecutar estos programa basta solamente con escribir su nombre.

2. ¿QUÉ SON LOS ARCHIVOS POR LOTES?

En esencia, se trata de un archivo de texto que contiene un listado de órdenes o comandos que se ejecutan uno detrás de otro. Todos los nombres de archivos de tratamientos por lotes deben de tener la extensión ".BAT".

Para crear un archivo ".BAT" se puede emplear el editor de texto "Block de Notas" de Windows con la única observación de que la extensión tiene que ser ".BAT". Como ya hemos dicho, la principal utilidad de un archivo ".BAT" o archivo de tratamiento por Lotes o simplemente un archivos por Lotes, es el de automatizar las tareas. Veamos un ejemplo sencillo:

Tengo un PC en que hay varios usuarios, entonces necesito borrar todos los días los archivos temporales que se encuentran en **C:\windows\temp**.

Mi tarea diaria es:

C:\>cd windows\temp Ingreso al directorio temporal

C:\windows\temp\>del *.tmp Borro los archivos temporales

Se puede automatizar esto mediante un archivo por Lotes, que lo construimos con algún editor, por ejemplo abrimos el Block de Notas de Windows y le agregamos los comandos anteriores.

Luego guardamos este archivo con el nombre de "ejemplo" y con su extensión ".BAT", ahora solo basta escribir "ejemplo" para que este archivo realice la tarea de borrar los archivos temporales.

Los archivos de tratamientos por lotes no deben de tener el mismo nombre que cualquier otra orden del S.O. o de cualquier programa de aplicación que el computador pueda tener. Si se crea de forma accidental un archivo por lotes que tenga el mismo nombre que una orden del S.O., el S.O. ejecutara siempre la orden y nunca el archivo por lotes. Todas las órdenes que el archivo por lotes contenga deberán ser escritas en cada línea del archivo, o sea una orden por línea.

3. ¿CÓMO FUNCIONA?

Primero, cuando se escribe el nombre del archivo por lotes, el S.O. realiza una revisión interna para comprobar si este nombre (o sea la cadena de caracteres que representan el nombre) que se ha escrito concuerda con algunas de las ordenes internas del S.O. . Si, efectivamente, concuerda, se ejecuta la orden. En caso contrario, el S.O. busca alguna orden externa o algún programa de aplicación (recordemos que todas las ordenes externa del S.O. y los programas de aplicación terminan con las extensión ".EXE" y ".COM"). Si se encuentra la orden o el programa, se ejecuta. De no ser así el S.O. ejecuta por último nuestro archivo por lotes. En resumen un archivo por lotes no debe de tener el nombre de alguna orden del S.O., ni el nombre de alguna aplicación o de cualquier otro programa. Si de forma accidental se crea un archivo por lotes que tenga el mismo nombre que una orden del S.O., el S.O. ejecutara siempre la orden y nunca el archivo por lotes.

4. CANCELACIÓN DE UN ARCHIVO POR LOTES

La forma más sencilla de cancelar la ejecución de archivo de tratamiento por lotes, una vez que este se ha inicializado su ejecución, es pulsando la tecla **<CTRL> + C**. Dependiendo del tipo de órdenes que constituyan el archivo por lotes, el S.O. puede esperar a que finalice la orden en curso en ese momento antes de detener la ejecución del archivo de tratamiento por lotes. El S.O. visualizara por pantalla el siguiente mensaje:

¿Finalizar el trabajo por lotes (S/N)?

Si realmente desea que se detenga la ejecución del archivo por lotes, pulse S; en caso contrario, pulse N y proseguirá su ejecución. Si alguna de las órdenes que constituyen el archivo por lotes se ha ejecutado sus efectos no son anulados. Por ejemplo si la primera orden dice que se borre un archivo, detener posteriormente un archivo por lotes no impide que se haya borrado el archivo.

5. ORDENES ESPECIALES PARA LOS ARCHIVOS POR LOTES

Dentro de un archivo por Lotes se pueden agregar órdenes especiales que le proporcionan un mayor control sobre la forma en que se interpreta o funciona el archivo mismo. Estas órdenes especiales permiten crear archivos de tratamientos por lotes que en realidad llegan a ser casi programas:

5.1 La Orden ECHO

La orden **ECHO** tiene dos usos. En primer lugar, se utiliza para controlar si el S.O. visualiza todas las órdenes por pantalla que contiene el archivo por lotes y otro uso es de, mostrar por pantalla algún mensaje.

ECHO está activada por defecto. Lo que quiere decir que el S.O. visualiza cada orden del archivo a medida que se ejecuta. Si ECHO se desactiva, no se visualizaran las ordenes de tratamiento por lotes, pero se mostrara cualquier resultado producido por las mismas.

La orden ECHO tiene la siguiente forma general.

ECHO on/off/mensaje

Para desactivar la orden ECHO, se escribe

ECHO OFF

Para activarlo, se escribe

ECHO ON

Por ejemplo creamos un archivo por lotes llamado E.BAT y escribimos las siguientes líneas:

```
ECHO OFF
```

```
VER
```

Cuando ejecutamos el archivo por lotes llamado E.BAT (tecleando E en el S.O.), se verá el resultado de la orden VER, pero no se verá que el S.O. ejecute realmente la orden. El resultado mostrado por este archivo por lotes tendrá este aspecto:

```
C:\>E
```

```
C:\>ECHO OFF
```

```
Microsoft Windows [Versión 10.0.10586]
```

```
C:\>
```

Ahora si se hubiese utilizado la orden ECHO ON, se visualiza cada orden a medida en se ejecuta. El resultado tiene ahora el siguiente aspecto:

```
C:\>ECHO ON
```

```
C:\>VER
```

```
Microsoft Windows [Versión 10.0.10586]
```

```
C:\>
```

Como se puede observar, esta vez C:\>VER apareció en pantalla. En resumen cuando la orden ECHO esta desactivado, tan solo aparece en la pantalla el resultado de la orden: Cuando ECHO está activado, el S.O. visualiza la ejecución de cada orden.

Cuando concluye la secuencia de órdenes del archivo de tratamiento por lotes, ECHO se activa automáticamente.

También se puede utilizar la ECHO para visualizar un mensaje en la pantalla. Para hacer esto, se escribe el mensaje simplemente después de la orden ECHO.

Por ejemplo, utilizaremos el archivo por lotes anterior, E.BAT, para una mejor comprensión.

ECHO OFF

ECHO Este mensaje saldrá por pantalla

ECHO Recordemos que la orden echo esta desactivada

ECHO y solo saldrá por pantalla estos mensajes y

ECHO el resultado de la orden VER

VER

El resultado sería:

C:\>E

C:\>ECHO OFF

Este mensaje saldrá por pantalla

Recordemos que la orden echo esta desactivada

y solo saldrá por pantalla estos mensajes y

los resultados de la orden VER

Microsoft Windows [Versión 10.0.10586]

C:\>

Hay que recordar que el mensaje aparecerá tanto si ECHO está activado o desactivo. Si se desea suprimir la visualización de tan solo algunas órdenes del archivo por lotes, puede resultar más sencillo colocar una @ delante de cada línea de orden del archivo por lotes.

5.2 La Orden PAUSE

Esta orden sirve para colocar una pausa en la ejecución del archivo de tratamiento por lotes, esta orden adopta la siguiente forma general:

PAUSE

Cuando se encuentra con una orden PAUSE, el S.O. visualiza el mensaje:

Presione cualquier tecla para continuar . . .

El S.O. esperara hasta que se pulse cualquier tecla para continuar la ejecución del archivo. Se puede cancelar el la ejecución de este archivo por lotes simplemente pulsando **<CTRL> + C**.

5.3 La Orden REM

Esta orden sirve para agregar un comentario dentro del archivo por lotes. Este comentario puede servirle al propio usuario o a cualquier persona que desee editar el archivo por lotes. Su formato es:

REM comentario

El comentario puede estar constituido por cualquier cadena de caracteres de una longitud entre 0 y 123 caracteres.

Esta orden también es útil para omitir una línea de orden en un archivo por lotes, en vez de borrarla se suprime colocando delante de la línea la orden REM.

Por ejemplo, hacemos un archivo por lotes llamado EJEM.BAT, que contiene lo siguiente:

ECHO OFF

REM Este es un comentario dentro del archivo por lotes

ECHO Este es un comentario que saldrá por pantalla

VER

REM VER

Y el resultado será:

```
C:\>EJEM
```

```
C:\>ECHO OFF
```

Este es un comentario que saldrá por pantalla

Microsoft Windows [Versión 10.0.10586]

```
C:\>
```

Como se puede observar el comentario de la orden REM no aparece en pantalla, pero si aparece el comentario de la orden ECHO. También se puede observar que la orden REM suprimió la línea de la orden VER, ya que se ejecutó una vez aunque estaba escrita para ejecutarse dos veces.

5.4 La Orden CALL

Algunas veces necesitará ejecutar otro archivo por lotes desde el interior de un archivo de tratamientos por lotes. La mejor forma de hacerlo, es mediante la orden CALL, cuya forma general es la siguiente:

CALL archivoporlote

En el que archivoporlote es el nombre del archivo que se quiere ejecutar.

Un archivo de tratamiento por lotes puede llamarse a sí mismo, pero se debe de asegurar de que alguna condición lo finalice ya que puede ejecutarse infinitamente.

5.5 Añadiendo Parámetros

A veces se puede crear un archivo de tratamiento por lotes que funcione ligeramente distinto según la forma en que se utilice.

Los archivos BATCH (o archivos por lotes, como también se le conocen) tienen la posibilidad de utilizar un indicador **%n**, donde "n" es un número

entre 0 y 9 para admitir parámetros reemplazables de entrada, o sea nos permiten crear archivos multiuso.

Podemos especificar hasta 10 parámetros de entrada (**%0 y %9**) de los cuales, el parámetro "%0" se sustituye siempre por el nombre del archivo por lotes.

Por ejemplo, creamos un archivo por lotes llamado MUEVE.BAT

COPY C:\%1 A:

DEL %1

CLS

Este archivo de tratamiento por lotes realiza tres funciones distintas; primero, copia los archivos que sean especificados por el parámetro %1 a la disquetera, luego borra en C:\ el/los archivos que copió y por último limpia la pantalla. Este archivo por lotes se puede usar de las siguientes formas:

C:\>MUEVE wperfect.exe

El cual moverá el archivo "wperfect.exe" a la unidad A, el parámetro "%1" será " wperfect.exe".

C:\>MUEVE *.exe

Aquí se moverán todos los archivos con la extensión ".exe" a la disquetera. En este caso el parámetro "%1" sería "*.exe", el cual puede ser cambiado por otro, según la utilidad que le dé el usuario.

Hasta ahora hemos estado tratando con archivos por lotes secuenciales, en los que cada línea se va ejecutando una tras otra. Pero esto no es lo habitual, puesto que es sabido que la verdadera potencia en un archivo BATCH reside en la posibilidad de efectuar saltos y bifurcaciones, permitiendo diversas secuencias lógicas.

5.6 La Orden IF

Con frecuencia resulta útil crear un archivo por lotes que haga cosas distintas dadas unas determinadas condiciones. Para llevar a cabo esto, el S.O. nos proporciona la orden IF destinada para los archivos de tratamiento por lotes, que adopta la siguiente forma general:

IF condición orden1 ELSE orden2

Aquí, condición se refiere a uno de los tres tipos de condiciones posibles y orden se refiere a cualquier orden del S.O. .

Si la condición resulta ser VERDADERA, se ejecuta la orden1 que sigue a la condición.

Orden1 puede ir seguido de la palabra clave **ELSE** y, a continuación, una orden2 que se ejecutará si la condición especificada es FALSA

Comprobar la igualdad de dos cadenas. Una cadena en el S.O. es simplemente una serie de caracteres. Se puede utilizar IF para comprobar la igualdad de dos cadenas utilizando la forma general:

IF cadena1==cadena2 orden

Si cadena1 es igual a cadena2, la condición es VERDADERA y, en caso contrario, es falsa.

Veamos un ejemplo sencillo:

ECHO OFF

IF ROJO==AMARILLO ECHO No saldrá nada en pantalla

IF ROJO==ROJO ECHO El color es ROJO

Este simple archivo lo que hace es imprimir por pantalla la sentencia "El color es ROJO" ya que comparó que la cadena ROJO es igual a la cadena ROJO y no es igual a la cadena AMARILLO. Naturalmente, comparar dos cadenas como las mostradas en el ejemplo anterior tiene muy poco valor práctico. Sin embargo, se puede utilizar esta característica para comparar los argumentos de dos líneas de órdenes. Modificando el ejemplo anterior, tenemos:

```
ECHO OFF
```

```
IF %1==AMARILLO ECHO El color es amarillo
```

```
IF %1== ROJO ECHO El color es rojo
```

Ahora para ejecutar este archivo, simplemente tipiamos su nombre (el que le hayamos puesto) más su parámetro, este archivo por lotes usa parámetros ya que como vemos tiene el parámetro %1.

Por ejemplo este archivo por lotes lo llamamos EJEMPLO.BAT, y lo ejecutamos en S.O. como "EJEMPLO ROJO", lo que nos mostrara en pantalla el mensaje "El color es rojo".

```
C:\>EJEMPLO ROJO
```

```
C:\>ECHO OFF
```

```
El color es rojo
```

```
C:\>
```

También podemos realizar distintos tipos de comparaciones usando el siguiente formato:

IF [/I] cadena1 comparación cadena2 orden

Donde comparación puede ser:

EQU – igual (equal)

NEQ - no igual

LSS - menor que

LEQ - menor que o igual

GTR - mayor que

GEQ - mayor que o igual

Si el modificador /I, si se especifica, realiza comparaciones de cadena que no distinguen entre mayúsculas y minúsculas. El modificador /I también puede usarse en la forma cadena1==cadena2 de IF.

Veamos algunos ejemplos:

@ECHO OFF

IF "23" LSS "12" (ECHO pues resulta que 23 es menor que 12) ELSE (ECHO pues resulta que 23 no es menor que 12)

@ECHO OFF

IF %1 EQU %2 (ECHO Los dos parámetros pasados son iguales) ELSE (IF %1 LSS %2 (ECHO El primer parámetro es menor) ELSE (ECHO El segundo parámetro es menor))

Comprobando si existe un archivo. Se puede comprobar si existe un archivo o un grupo de archivos utilizando la condición EXIST de la orden IF, que tiene la siguiente forma general:

IF EXIST **nombredearchivo orden**

En el que nombredearchivo es el nombre del archivo que se está buscando. El nombredearchivo puede incluir tanto un especificador de unidad como un nombre de camino.

Comprobando la existencia de errores. Un programa de aplicación puede utilizar una variable interna del S.O. que indica que el programa terminó normalmente o no, debido a un error.

Para facilitar la explicación, vamos a llamar a esta variable, variable error. Si un programa termina de forma normal, pone la variable de error a cero, indicando que todo salió bien. Si termina debido a un error pone en la variable de error un número mayor que cero. Si un programa no utiliza

realmente el valor de la variable error, éste es por defecto, cero. El S.O. permite comprobar esta variable de error usando la condición ERRORLEVEL (nivel de error) de la orden IF, que adopta la siguiente forma general:

IF ERRORLEVEL n orden

En el que n es un número mayor o igual que cero y representa el número de error escrito por el programa de aplicación. Si el valor de la variable error es mayor o igual que n, la condición es VERDADERA.

Francamente, el uso de ERRORLEVEL es algo complicado, y la orden es utilizada con mayor frecuencia por programadores.

El uso de NOT. Se puede colocar delante de la condición IF la palabra NOT, la cual, entonces, cambiara completamente el resultado de la condición. Por ejemplo, si

EXIST (Archivocualquiera) es VERDADERA

NOT EXIST (Archivocualquiera) es FALSA

Para comprender el uso de NOT, supongamos que se tiene un programa de aplicación que requiere que el archivo INFO.DOC esté presente. Se podría usar la siguiente orden en un archivo por lotes para buscar el archivo antes de ejecutar el programa.

IF NOT EXIST INFO.DOC PAUSE Mo existe el archivo INFO.DOC

5.7 La Orden GOTO

La orden de tratamientos por lotes **GOTO** se emplea para indicarle al S.O. que ejecute las órdenes de un archivo por lotes en un orden no secuencial. La forma general de GOTO es:

GOTO etiqueta

En el que etiqueta es una etiqueta definida en alguna otra parte del mismo archivo por lotes. Cuando se ejecuta GOTO, el S.O. salta a la etiqueta especificada y comienza a ejecutar las órdenes a partir de ese punto. Usando GOTO, se puede hacer que la ejecución del archivo BATCH salte hacia delante o hacia atrás en sus líneas de órdenes.

Por ejemplo, creamos un archivo por lotes llamado GOTOP.BAT y escribimos en él las siguientes líneas:

ECHO OFF

IF %1==ROJO GOTO CROJO

IF %1==AZUL GOTO CAZUL

:CROJO

ECHO Eligió el color ROJO

DIR

GOTO FIN

:CAZUL

ECHO Te gusta el color AZUL

:FIN

ECHO Termino tu archivo por lotes

Este archivo por lotes tiene parámetros y dependiendo del parámetro que le ingresemos nos dará un resultado distinto en ROJO y AZUL.

Como se puede observar del ejemplo anterior, todas las etiquetas deben de comenzar con dos puntos.

Se puede utilizar la orden GOTO junto con la orden IF para crear bloques de órdenes que se ejecutaran tan solo si la condición IF es VERDADERA.

Cuando se emplea la orden GOTO, en las etiquetas que contienen ordenes hay que colocar dentro de ellas otra orden GOTO que salte a otro bloque de ordenes (como en el ejemplo GOTO FIN), porque si no se ejecutara las ordenes GOTO que siguen del siguiente bloque.

5.8 Repetición de Ordenes con FOR/DO

Se puede repetir una serie de órdenes que usen distintos argumentos utilizando la orden FOR, que adopta la siguiente forma general:

FOR %%var IN (listas argumentos) DO orden

FOR %%var IN (lista argumentos) DO (

comando

comando

...

)

Observa que la variable "var" va precedida por **dos símbolos de "%"**. Además **el nombre de la variable debe ser UNA SOLA LETRA** (p.ej: %%n, %%i, %%j , etc) que irá tomando los valores de la lista de argumentos. Los argumentos deben estar separados por espacios en blanco. FOR repetirá la orden tantas veces como argumentos haya. Cada vez que FOR repita, var será sustituido por argumento desplazándose de izquierda a derecha en la lista.

EJEMPLO1:

Para que la variable del FOR vaya tomando distintos valores dentro de una lista determinada:

FOR %%X IN (%*) DO (

ECHO %%X

)

Esto sacará por pantalla las palabras que están entre paréntesis, separadas línea a línea.

EJEMPLO2

Si queremos recorrer una lista de archivos de un determinado directorio (solo archivos, no directorios):

```
FOR %%F IN (*) DO (
```

```
ECHO %%F
```

```
)
```

y si queremos mover a la papelera de reciclaje algunos archivos con determinadas extensiones:

```
FOR %%F IN (*.JPG, *.MP3, *.BMP) DO (
```

```
MOVE %%F C:\RECYCLER
```

```
)
```

Veamos un ejemplo que muestra en pantalla los tres primeros argumentos de la línea de órdenes con la que se llame.

```
ECHO OFF
```

```
FOR %%i IN (%1 %2 %3) DO ECHO %%i
```

Si ejecutamos SIMPFOR JPG GIF BMP el resultado Sera:

JPG

GIF

BMP

Se puede usar FOR, para ejecutar una lista de órdenes colocándolas en la lista de argumentos. Por ejemplo, la siguiente línea de ordenes hará lo siguiente: limpiara la pantalla, luego mostrara los directorios y por último hará un chequeo del disco, obviamente usando comandos del S.O. .

```
FOR %%C IN (cls dir chkdsk) DO %%C
```


Una orden FOR no se puede usar para ejecutar otra orden FOR, o sea no se puede usar como objeto de DO.

5.9 La Orden SHIFT

Como ya sabemos, tan solo hay 10 parámetros sustituibles, del %0 al %9. Se puede emplear la orden SHIFT para conseguir el acceso a más de 10 argumentos de la línea de órdenes.

Cada vez que se ejecuta SHIFT, el contenido de los parámetros reemplazables es desplazado hacia la izquierda una posición, lo que había en %0 se pierde y el nuevo argumento, si existe, entra en %9, ocupando en %0 lo que había en %1, el %1 tomaría lo que había en %2, etc.

Veamos un ejemplo sencillo, creando un archivo BATCH llamado DESPLA.BAT que contengan las siguientes órdenes:

ECHO OFF

ECHO %0 %1 %2 %3

SHIFT

ECHO %0 %1 %2 %3

Si la ejecutamos con los siguientes argumentos MOV PRI SEG TER, tendremos:

C:\>DESPLA MOV PRI SEG TER

C:\>ECHO OFF

MOV PRI SEG TER

PRI SEG TER

Otro ejemplo, creamos un archivo de tratamiento por lotes llamado MOSTRAR.BAT, lo que contendrá:

@ECHO OFF

ECHO El parámetro 1 es %1

ECHO El parámetro 2 es %2

ECHO El parámetro 3 es %3

SHIFT

ECHO Ahora el parámetro 1 es %0

ECHO Ahora el parámetro 2 es %1

ECHO Ahora el parámetro 3 es %2

ECHO Por último el parámetro 4 es %3

PAUSE

Si tecleamos en el S.O., C:\>MOSTRAR A B C D, en pantalla aparecerá:

El parámetro 1 es A

El parámetro 2 es B

El parámetro 3 es C

Ahora el parámetro 1 es A

Ahora el parámetro 2 es B

Ahora el parámetro 3 es C

Por último el parámetro 4 es D

Lo más importante que hay que recordar en torno a los argumentos que se usen en un archivo por lotes, es que deben estar separados por espacios en blanco, ya que S.O. no reconoce ningún otro carácter como separador.

5.10 Uso de variables

En los archivos BAT se pueden utilizar variables para luego poder recuperar su valor introduciendo ésta entre signos de porcentajes (**%NombreVariable%**).

Para establecer una variable, se utiliza el comando **SET**.

Set: Da valor a una variable general o de entorno.

Ejemplo:

```
set nombre=Luis
```

Parámetros:

set /a Da valor a una variable utilizando operaciones aritméticas

Ejemplo: `set /a número=2+2` Lo que devolvería el valor 4

set /p Permite la entrada de datos por parte del usuario (prompt), el valor introducido por el mismo define la variable.

Ejemplo:

```
set /p nombre=Introduce tu nombre
```

Esto definiría la variable `%nombre%` con el valor introducido por el usuario.

Nota: Todo lo que vaya después del signo de = en un `set /p` se verá en pantalla.

Una vez creada la variable, se invoca con **%variable%**.

Por ejemplo:

```
SET var1="C:\archivos de programa"
```

Si desde C: pones:

```
C:> cd %var1%
```

El resultado es como poner:

```
C:> cd "C:\archivos de programa"
```

Puedes crear una variable con los datos que quieras, pero para moverte entre directorios te recomiendo que utilices comillas (" ").

Para almacenar un número en una variable, utiliza **SET /a** .

Ejemplo:

```
SET /a numero1=50
```

```
SET /a numero2=25
```

Para realizar operaciones, puedes utilizar lo siguiente:

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

Para comprobarlo, pon:

```
ECHO Suma de %numero1% + %numero2% = %suma%
```

El resultado debería ser:

```
Suma de 50 + 25 = 75
```

Ejemplo de creación de un menú:

```
@ECHO OFF
```

```
CLS
```

```
:MENU
```

```
ECHO 1) PRIMERA OPCION
```

```
ECHO 2) SEGUNDA OPCION
```

```
ECHO 3) TERCERA OPCION
```

```
ECHO 4) CUARTA OPCION
```

```
ECHO 5) SALIR
```

```
:BUCLE
```

set /p Opcion=Introduce Opción:

IF %Opcion%==1 (

ECHO Has seleccionado la opción 1

PAUSE

CLS

GOTO MENU

)

IF %Opcion%==2 (

ECHO Has seleccionado la opción 2

PAUSE

CLS

GOTO MENU

)

IF %Opcion%==3 (

ECHO Has seleccionado la opción 3

PAUSE

CLS

GOTO MENU

)

IF %Opcion%==4 (

ECHO Has seleccionado la opción 4

PAUSE

CLS

GOTO MENU

)

IF %Opcion%==5 (

ECHO Adios

GOTO FIN

)

GOTO BUCLE

:FIN