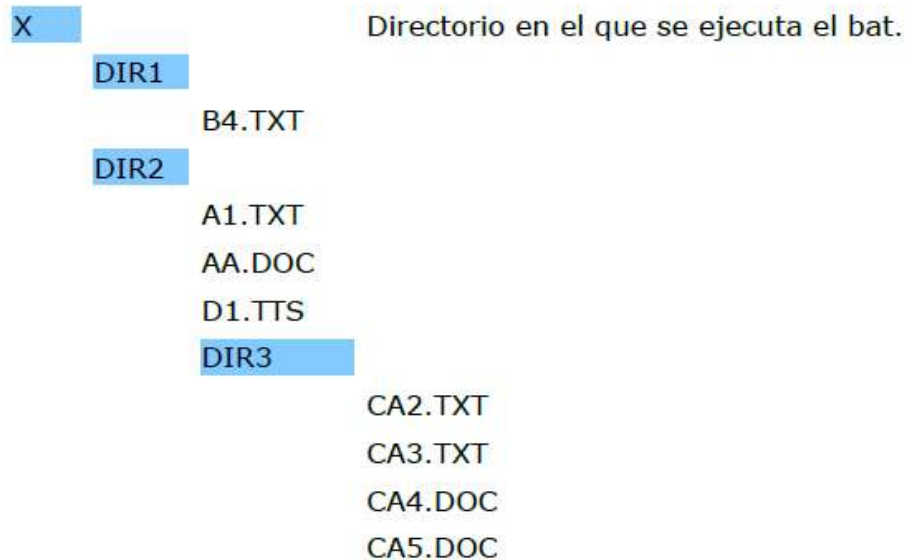


EJERCICIO 1. Dada la siguiente estructura de directorios:



Los elementos con fondo azul son directorios

Desarrolla un bat que genere la estructura indicada en la carpeta X. Además, antes de crear la estructura nos deberá preguntar si queremos loguear su creación. En caso de que respondamos que sí, creará un fichero llamado LOG.TXT en la carpeta X en el cual almacenará la siguiente información:

Iniciando creación de estructura

Hora de inicio:

XXXX (donde XXXX es la hora a la que se inicio la creación de la estructura de directorios)

Para cada elemento creado de la estructura de directorios:

Comando ejecutado

Hora de ejecución:

XXXX (donde XXXX es la hora a la que se ejecuto el comando)

Creación de estructura finalizada.

Hora de finalización:

XXXX (donde XXXX es la hora a la que finalizo la creación de la estructura de directorios)

@echo off

set /p pregunta=Quieres loguear la creacion?

set horaInicio=time /t

md DIR1

set horaComDIR1=time /t

echo.>DIR1/B4.txt

set horaComB4=time /t

md DIR2

set horaComDIR2=time /t

echo.>DIR2/A1.txt

set horaComA1=time /t

echo.>DIR2/AA.doc

set horaComAA=time /t

echo.>DIR2/D1.tts

set horaComD1=time /t

md DIR2\DIR3

set horaComDIR3=time /t

echo.>DIR2/DIR3/CA2.txt

set horaComCA2=time /t

echo.>DIR2/DIR3/CA3.txt

set horaComCA3=time /t

echo.>DIR2/DIR3/CA4.doc

set horaComCA4=time /t

echo.>DIR2/DIR3/CA5.doc

set horaComCA5=time /t

if %pregunta%==si goto loguear

set horaFin=time /t

goto fin

:loguear

echo Iniciando creacion de estructura >> LOG.txt

echo Hora de inicio: >> LOG.txt

%horaInicio% >> LOG.txt

```
echo Comando ejecutado: "md DIR1" >> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComDIR1% >> LOG.txt
```

```
echo Comando ejecutado: "echo.>DIR1/B4.txt" >> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComB4% >> LOG.txt
```

```
echo Comando ejecutado: "md DIR2">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComDIR2%>> LOG.txt
```

```
echo Comando ejecutado: "echo.>DIR2/A1.txt">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComA1%>> LOG.txt
```

```
echo Comando ejecutado: "echo.>DIR2/AA.doc">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComAA%>> LOG.txt
```

```
echo Comando ejecutado: "echo.>DIR2/D1.tts">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComD1%>> LOG.txt
```

```
echo Comando ejecutado: "md DIR2\DIR3">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComDIR3%>> LOG.txt
```

```
echo Comando ejecutado: "echo.>DIR2/DIR3/CA2.txt">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComCA2%>> LOG.txt
```

```
echo Comando ejecutado: "echo.>DIR2/DIR3/CA3.txt">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComCA3%>> LOG.txt
```

```
echo Comando ejecutado: "echo.>DIR2/DIR3/CA4.doc">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComCA4%>> LOG.txt
```

```
echo Comando ejecutado: "echo.>DIR2/DIR3/CA5.doc">> LOG.txt
echo Hora de ejecución:>> LOG.txt
%horaComCA5%>> LOG.txt
```

```
echo Creación de estructura finalizada >> LOG.txt
echo Hora de finalización.: >> LOG.txt
%horaFin% >> LOG.txt
:fin
echo Fin del programa
```

EJERCICIO 2. Crea un fichero bat que reciba 5 parámetros. Debe comprobar que hay 5 y sólo 5. En caso de que no se cumpla la premisa anterior deberá mostrar un mensaje de error. En caso de que haya 5 deberá mostrar un mensaje de éxito.

```
@echo off
```

```
if "%5"==" " goto pocos
if not "%6"==" " goto muchos
```

```
echo Numero de parametros correcto
goto fin
```

```
:pocos
echo El numero de parametros debe ser mayor
goto fin
```

```
:muchos
echo El numero de parametros debe ser menor
```

```
:fin
echo Fin del programa
```

EJERCICIO 3. Crea un fichero bat que reciba N parámetros. Debe mostrarlos con el siguiente formato:

```
PARAMETRO 1. VALOR: XXX (donde XXX es el valor del parámetro 1)
PARAMETRO 2. VALOR: XXX (donde XXX es el valor del parámetro 2)
PARAMETRO 3. VALOR: XXX (donde XXX es el valor del parámetro 3)
...
PARAMETRO N. VALOR: XXX (donde XXX es el valor del parámetro N)
```

```
@echo off
```

```
setlocal enabledelayedexpansion
set contador=1
```

```
for %%i in (%) do (
echo PARAMETRO !contador!. VALOR: %%i
```

```
set /a contador=!contador!+1
)
echo fin del programa
```

EJERCICIO 4. Crea un fichero bat que reciba una cadena de texto como parámetro. El programa debe validar que el parámetro recibido no es vacío (en caso de que sea vacío mensaje de error y fin del script). A continuación nos pedirá cadenas de texto por pantalla hasta que acertemos la que pasamos como parámetro. En el momento que acertemos, el programa nos indicará que hemos acertado, cuantos intentos hemos necesitado y finalizará. Si en algún momento introducimos una cadena vacía, deberá indicárnoslo y pedirnos otra cadena de nuevo para continuar con el juego (una cadena vacía no se considerara como un intento de acierto).

```
@echo off
setlocal enabledelayedexpansion

set contador=0
if "%1"=="" goto error

:fallo

set /a contador=contador+1

:retornoVacio
set cadena=
set /p cadena=Introduce un texto:
if %*==!cadena! goto acierto

if "!cadena!"=="" goto vacio
goto fallo

:acierto
echo Has acertado
echo Has necesitado %contador% intento/s
goto fin

:error
echo Error:el texto esta vacio
goto fin

:vacio
echo La cadena esta vacia
goto retornoVacio

:fin
```

EJERCICIO 5. Crear un fichero bat que esté preparado para recibir 3 parámetros. En caso de que se invoque con más o con menos parámetros, se mostrará un mensaje de error al respecto y el programa finalizará. En caso de que el número de parámetros sea correcto se deberá validar que se cumplen las siguientes condiciones:

- El valor del primer parámetro es /a
- El valor del segundo parámetro es /b
- El valor del tercer parámetro es ABC
-

En caso de que no se cumpla alguna de las condiciones anteriores, se mostrará un mensaje de error y el programa finalizará.

```
@echo off
```

```
if "%3"==" " goto pocos
```

```
if not "%4"==" " goto muchos
```

```
goto continuar
```

```
:pocos
```

```
echo El numero de parametros debe ser mayor
```

```
goto fin
```

```
:muchos
```

```
echo El numero de parametros debe ser menor
```

```
goto fin
```

```
:continuar
```

```
if not "%1"==" /a" goto condicion1
```

```
goto evaluar2
```

```
:condicion1
```

```
echo El valor del primer parametro debe ser /a
```

```
:evaluar2
```

```
if not "%2"==" /b" goto condicion2
```

```
goto evaluar3
```

```
:condicion2
```

```
echo El valor del segundo parametro debe ser /b
```

```
:evaluar3
```

```
if not "%3"=="ABC" goto condicion3
```

```
goto validar
```

```
:condicion3
```

```
echo El valor del tercer parametro debe ser ABC
```

```
goto fin
```

```
:parametro1Validado
```

```
if "%2"==" /b" goto parametro2Validado
```

```

goto fin

:parametro2Validado
if "%3"=="ABC" goto parametro3Validado
goto fin

:parametro3Validado
echo Parametros validados
goto fin

:validar
if "%1"=="a" goto parametro1Validado

:fin
echo Fin del programa

```

EJERCICIO 6. Crear un fichero bat que este preparado para recibir 3 parámetros. En caso de que se invoque con más o con menos parámetros, se mostrará un mensaje de error al respecto y el programa finalizará. En caso de que el número de parámetros sea correcto se deberán realizar las siguientes acciones en función al valor del tercer parámetro:

- Si el tercer parámetro vale COPIAR, se copiará el fichero indicado como primer parámetro al directorio indicado como segundo parámetro. Antes de realizar la copia se deberá comprobar que el fichero y el directorio existen. Si no existieran mensaje de error y fin de programa.
- Si el tercer parámetro vale MOVER, se moverá el fichero indicado como primer parámetro al directorio indicado como segundo parámetro. Antes de mover el archivo se deberá comprobar que el fichero y el directorio existen. Si no existieran mensaje de error y fin de programa.
- Si el tercer parámetro tiene un valor diferente a COPIAR o a MOVER se mostrará un mensaje de error indicando que la sintaxis no es correcta.

Tanto al copiar como al mover, el fichero bat no nos debe pedir confirmación para nada.

```

@echo off

if "%3"=="" goto pocos
if not "%4"=="" goto muchos

goto continuar

:pocos
echo El numero de parametros debe ser mayor
goto fin

:muchos
echo El numero de parametros debe ser menor
goto fin

```

:continuar

if "%3"=="COPIAR" goto copiar

if "%3"=="MOVER" goto mover

goto error

:copiar

if not exist "%1" goto noExisteFichero

if not exist "%2" goto noExisteDirectorio

copy "%1" "%2" /y

goto fin

:mover

if not exist "%1" goto noExisteFichero

if not exist "%2" goto noExisteDirectorio

move "%1" "%2"

goto fin

:noExisteFichero

echo No existe el fichero

goto fin

:noExisteDirectorio

echo No existe el directorio

goto fin

:error

echo La sintaxis no es correcta

:fin

echo Fin del programa

EJERCICIO 7. Desarrolla un fichero bat que reciba N parámetros:

- El primer parámetro será una carpeta de ORIGEN.
- El segundo parámetro será una carpeta de DESTINO.
- El tercer parámetro será la OPERACION a realizar.
- Los parámetros existentes entre el tercero y el último serán extensiones de archivos (txt, doc, zip, etc...).

El fichero bat validará que al menos hay 4 parámetros. En caso de error finalizará.

El fichero bat validará que ORIGEN existe. En caso de que no exista finalizará.

El fichero bat validará que DESTINO existe. En caso de que no exista finalizará.

El fichero bat validará que OPERACION vale 1,2,3 o 4. En caso de que no lo haga el fichero bat finalizará.

Acciones a realizar en función del valor de operación:

OPERACION=1

- Copia el contenido de la carpeta ORIGEN a la carpeta DESTINO.

OPERACION=2

- Muestra los atributos de todos los archivos existentes en la carpeta DESTINO.

OPERACION=3

- Muestra todas las carpetas y subcarpetas (ficheros no) existentes en la carpeta ORIGEN.

OPERACION=4

- Muestra el número total de archivos (carpetas no) con las extensiones pasadas como parámetros existentes en la carpeta ORIGEN y en la carpeta DESTINO

@echo off

```
if "%4"==" " goto pocos
set contador=0
goto continuar
```

```
:pocos
echo El número de parámetros debe ser mayor
goto fin
```

```
:continuar
```

```
if not exist "%1" goto origenNoExiste
if not exist "%2" goto destinoNoExiste
if "%3"=="1" goto 1
if "%3"=="2" goto 2
if "%3"=="3" goto 3
if "%3"=="4" goto 4
```

goto fin

:1

xcopy /e "%1" "%2" /y

goto fin

:2

attrib "%2*"

goto fin

:3

dir "%1" /a:d /s

goto fin

:4

for %%i in (%*) do (

for %%i in ("%1*.*", "%2*.*") do (

set /a contador=contador+1

)

)

echo %contador%

goto fin

:origenNoExiste

echo Origen no existe

goto fin

:destinoNoExiste

echo Destino no existe

goto fin

:error

echo La sintaxis no es correcta

:fin

echo Fin del programa

UD4.PRACTICA4

EJERCICIO 1. Desarrolla un fichero bat que reciba 9 parámetros. Validará que son 9. En caso de que no haya 9 finalizará con un mensaje de error. Al noveno parámetro lo denominaremos PATRON. El fichero bat debe indicar cuantas veces aparece el PATRON en los 8 primeros parámetros pasados.

```
@echo off
setlocal enabledelayedexpansion

shift
if "%8"==" " goto pocos
if not "%9"==" " goto muchos

set patron=%8
echo Numero de parametros correcto
goto continuar

:pocos
echo Los parametros son menos que 9
goto fin

:muchos
echo Los parametros son mas que 9
goto fin

contador=0

:continuar

for %%i in (%*) do (
if %patron%==%%i (
set /a contador=!contador!+1))
set /a contador=!contador!-1
echo Numero de veces que aparece patron: %contador%

:fin
echo Fin de programa
```

EJERCICIO 2. Desarrolla un fichero bat similar al desarrollado en el ejercicio 1. La diferencia es que podrá recibir un número de parámetros indeterminado y el PATRON será solicitado al usuario al iniciar la ejecución del fichero bat.

```
@echo off
setlocal enabledelayedexpansion

set patron=

set /p patron=Define un patron:
set contador=0

for %%i in (*) do (
if %patron%==%%i (
set /a contador=!contador!+1))

echo Numero de veces que aparece patron: %contador%

:fin
echo Fin de programa
```

EJERCICIO 3. Desarrolla un fichero bat que reciba 2 parámetros. Validará que son 2. En caso de que no haya 2 finalizará con un mensaje de error. El primer parámetro será una extensión de archivos. El segundo parámetro será un directorio. Comprobar que el directorio pasado como parámetro existe. De no existir, el fichero bat finalizará. El fichero bat debe de indicar cuantos ficheros de la extensión dada como primer parámetro existen en la carpeta indicada como segundo parámetro o en sus subcarpetas.

```
@echo off
setlocal enabledelayedexpansion
set contador=0

if "%2"=="" goto pocos
if not "%3"=="" goto muchos

if not exist "%2" goto noExiste

for /r %%i in (*.%1) do (
set /a contador=!contador!+1
)

echo Numero de ficheros que hay con la extension dada: %contador%

goto fin

:pocos
echo Debe de haber mas parametros
goto fin

:muchos
echo Debe de haber menos paremetros
```

```
goto fin
```

```
:noExiste
```

```
echo El fichero no existe
```

```
:fin
```

```
echo Fin del programa
```

EJERCICIO 4. Desarrolla un fichero bat que reciba N parámetros. Si recibe 0 ó 1 parámetros el fichero bat finalizará mostrando un mensaje de error.

En caso de que reciba 2 o más parámetros, el fichero bat indicará cuantas veces aparece el ultimo parámetro recibido en la lista de parámetros que lo preceden.

P.j: supongamos que el fichero bat se llama EJ2.BAT

COMANDO EJECUTADO	RESULTADO
EJ2 a a hola c f c a	2
EJ2 a a hola c f c a hola	1
EJ2 a a hola c f c a adios	0

En caso de que el programa funcione pero no la haga para N parámetros el ejercicio será puntuado con 0 puntos.

```
@echo off
```

```
setlocal enabledelayedexpansion
```

```
set contador=0
```

```
if "%2"==" " goto pocos
```

```
rem ultimo: variable donde almaceno el último parámetro
```

```
set ultimo=
```

```
for %%i in (*) do (
```

```
set ultimo=%%i
```

```
)
```

```
rem for para contar las veces que aparece último
```

```
for %%i in (*) do (
```

```
if %%i==%ultimo% (
```

```
set /a contador=!contador!+1
```

```
)
```

```
)
```

```
set /a contador=!contador!-1
```

```
echo Numero de veces que aparece el ultimo parametro: %contador%
```

goto fin

:pocos

echo Debe recibir 2 o mas parametros

:fin

echo Fin del programa

EJERCICIO 5. Crear un fichero BAT que gestione los usuarios del sistema operativo. El fichero BAT analizará las líneas del fichero USUARIOS.TXT y actuará en consecuencia. El formato de los registros del archivo USUARIOS.TXT es el siguiente:

USUARIO:OPERACION:ACTIVO:GRUPO:CLAVE

Descripción de los campos:

- USUARIO: nombre de usuario.
- OPERACION: Puede tomar uno de los siguientes valores:0,1. Si vale 0 indica que se quiere eliminar el usuario. Si vale 1 indica que se quiere crear el usuario. En caso de que el campo OPERACION valga 0, el registro no contendrá mas campos. En caso de que el campo OPERACION valga 1, el registro además contendrá los campos ACTIVO, GRUPO y CLAVE
- ACTIVO: indica si el usuario a crear estará o no habilitado. Si vale 0 el usuario no estará habilitado. Si vale 1 el usuario estará habilitado.
- GRUPO:es el nombre del grupo al que pertenecerá el usuario a crear.
- CLAVE: es la contraseña que se le asignará al usuario a crear.

Para cada uno de los registros del fichero USUARIOS.TXT cuyo campo operación valga 0 se comprobará que existe el usuario asociado. Para aquellos usuarios que existan se procederá a su eliminación. Además se almacenará en un archivo de log llamado LOG.TXT información acerca de los usuarios que se eliminaron y acerca de los que no pudieron eliminarse por no existir.

Para cada uno de los registros del fichero USUARIOS.TXT cuyo campo operación valga 1 se comprobará que el usuario asociado no existe. En caso de que ya exista se realizará una entrada en el archivo de log indicándolo. En caso de que el usuario no exista se comprobará que el grupo asociado ya existe. En caso de que no exista se realizará una entrada en el archivo de log indicándolo. En caso de que el grupo sí que exista se dará de alta un usuario con las características definidas en el archivo USUARIOS.TXT. La creación del usuario también se logueará.

Ejemplo de fichero USUARIOS.TXT:

ana:0

luis:0

juan:1:1:g1:aaaaaa123123

luis:1:0:g1:awerggaa123123

pedro:0

marta:1:0:g3:asdfaaaaa12sdf3

maria:1:1:g1:a34734aaaaa123

```

@echo off

setlocal enabledelayedexpansion

for /f "delims=: tokens=1,2,3,4,5" %%i in (USUARIOS.txt) do (

rem ELIMINACION DE USUARIO

if %%j==0 (
    net user %%i
    if "!ERRORLEVEL!"=="0" (
        net user %%i /DELETE
        echo El usuario %%i ha sido eliminado con exito >> LOG.TXT
        echo El usuario %%i ha sido eliminado con exito
    ) else (
        echo El usuario %%i no existe.
        echo El usuario %%i no se pudo eliminar porque no existe >> LOG.TXT
    )
)

rem CREACION DE USUARIO

if %%j==1 (
    rem SI EL USUARIO YA EXISTE
    net user %%i
    if "!ERRORLEVEL!"=="0" (
        echo El usuario %%i ya existe
        echo El usuario %%i ya existe >> LOG.TXT
    ) else (
        rem SI EL USUARIO NO EXISTE, CREACION NO HABILITADA
        if %%k==0 (
            net user %%i %%m /ADD /ACTIVE:NO
            echo Usuario %%i creado >> LOG.TXT
            echo Usuario %%i creado
            net localgroup %%l
            if "!ERRORLEVEL!"=="0" (
                net localgroup %%l /add %%i
                echo Usuario %%i incluido en el grupo %%l
                echo Usuario %%i incluido en el grupo %%l >> LOG.TXT
            ) else (
                echo El grupo %%l no existe
                echo El grupo %%l no existe >> LOG.TXT
            )
        )

        rem SI EL USUARIO NO EXISTE, CREACION HABILITADA
        if %%k==1 (
            net user %%i %%m /ADD /ACTIVE:YES
            echo Usuario %%i creado >> LOG.TXT

```

```
echo Usuario %%i creado
net localgroup %%l
    if "!ERRORLEVEL!"=="0" (
        net localgroup %%l /add %%i
        echo Usuario %%i incluido en el grupo %%l
        echo Usuario %%i incluido en el grupo %%l >> LOG.TXT
    ) else (
        echo El grupo %%l no existe
        echo El grupo %%l no existe >> LOG.TXT
```

```
)
```

```
)
)
)
)
)
```