

# Sistemas Operativos. Comandos 11

[Información sobre el documento.....1](#)

[1. - Introducción.....1](#)

[2. - Desarrollo de la actividad.....1](#)

[Usuario por parámetro.....2](#)

[Comprobando el valor de vuelta de la ejecución de un comando.....5](#)

[¿Cómo comprobar que el usuario tiene privilegios de administrador?.....7](#)

## Información sobre el documento

El objetivo de este documento es enseñar, compartir conocimientos para facilitar el aprendizaje. Este documento es mejorable, y será actualizado si es preciso.

Este documento puede ser utilizado para uso personal, no comercial, como se presenta, respetando una serie de **condiciones de uso basadas en Licencia Creative Commons**, como aparece en el logotipo, que se describe a continuación



- El **documento se proporciona como está**, por tanto *no se pueden realizar modificaciones*, ni en el **formato** ni en el **contenido**, ni trabajos derivados, sin la autorización expresa del autor.
- Se debe **mencionar al autor** del mismo, por supuesto sin modificar los enlaces o imágenes introducidas por éste en el documento.
- **No se puede utilizar** este material **con fines lucrativos, comerciales o cualquier uso que pueda proporcionar, directa o indirectamente, un beneficio económico de terceros**, sin la autorización expresa del autor.
- Si se desea publicar el documento en algún sitio web, se debe hacer a través de un **enlace al documento en el sitio del autor** ([www.educatica.es](http://www.educatica.es) o [cursos.educatica.es](http://cursos.educatica.es))

Si se desea publicar como un recurso dentro de un sitio web, sin utilizar un enlace al material en el sitio web del autor, se debe solicitar autorización expresa y, en cualquier caso, referenciar el sitio web del autor ([www.educatica.es](http://www.educatica.es)).

Espero sea de provecho ;)

## 1. -Introducción

El objetivo de esta actividad es refinar procesos por lotes dados para facilitar su comprensión y posterior mantenimiento y modificación.

La idea es sacar partido de las herramientas disponibles para mejorar nuestros procesos por lotes. En este caso, trabajaremos con parámetros, comprobaciones con el comando IF, etiquetas y saltos incondicionales.

## 2. -Desarrollo de la actividad

Partiendo del proceso por lotes imagenesBackup.bat refinado en la actividad 10, vamos a analizar el código y modificarlo para hacerlo más flexible. El objetivo principal es conseguir que se puedan realizar copias de seguridad de ficheros de imágenes de otros usuarios del sistema que no tengan que ser el usuario actual con el que estamos trabajando.

Para ello, necesitamos obtener el nombre de la cuenta de usuario para la que realizaremos la copia de seguridad por parámetro.



## Usuario por parámetro

Vamos a modificar el script para que ahora se pase el nombre del usuario por parámetro. Para ello usamos los parámetros dentro de los scripts, accediendo a ellos a través de las variables de parámetro de %0, que indica el nombre del script, y %1 a %9, que almacenan los parámetros que se han pasado.

```

1 @echo off
2 REM Copia de seguridad de las imágenes.
3 REM Versión con:
4 REM     - Nombre de usuario proporcionado por parámetro.
5 REM     - Comprobación de directorio de perfil de usuario
6 REM
7 REM Autor: Javier Pedrero Martín
8 REM Fecha: 29/01/2019
9 REM Comprobamos si se pasa por parámetro nombre de usuario y si existe su perfil
10 IF "%1"==" " GOTO NO_PARAM
11 SET user=%1
12 SET userDir="C:\Users\%user%"
13
14 IF NOT EXIST %userDir% goto NO_PROFILE
15
16 REM Variables que almacenan directorio de origen, destino y fichero de inf.
17 SET dirOrigen="%userDir%"
18 SET dirDestino="C:\Backup\%user%\imagenes"
19 SET infoFile=%dirDestino%\listado.txt
20
21 REM realizamos las comprobaciones previas
22 IF NOT EXIST %dirDestino% MKDIR %dirDestino%
23
24 REM Almacenamos en el fichero de información el listado de ficheros
25 REM de imágenes a copiar y la información usuario, fecha y hora.
26 SET extList=JPG BMP PNG GIF
27 FOR %e IN (%extList%) DO DIR /S %dirOrigen%\*.*%e >> %infoFile%
28 echo %USERNAME% %DATE% %TIME% >> %infoFile%
29
30 REM Realizamos la copia
31 FOR %e IN (%extList%) DO XCOPY /S %dirOrigen%\*.*%e %dirDestino%
32 echo "Copia.. OK"
33 GOTO FIN
34
35 :NO_PARAM
36 echo Se debe proporcionar por parámetro el nombre del usuario
37 echo para el que hacer la copia de seguridad
38 GOTO FIN
39
40 :NO_PROFILE
41 echo No existe el directorio %UserDir%
42 echo Puede que el usuario %user% no exista.
43 net user %user%
44
45 :FIN

```

En la línea 10 se comprueba si se ha pasado algún valor en el parámetro 1. Para ello, comparamos la cadena de texto "%1" con la cadena "". Ambas cadenas serán iguales solo cuando %1 no contenga ningún valor y el usuario haya ejecutado el script sin pasar ningún nombre de usuario.

```
10 IF "%1"==" " GOTO NO_PARAM
11 SET user=%1
12 SET userDir="C:\Users\%user%"
13
14 IF NOT EXIST %userDir% goto NO_PROFILE
```

Si ese es el caso, el script realiza un salto incondicional a la etiqueta NO\_PARAM, que está en la línea 35. Es decir, saltará todas las líneas posteriores sin ejecutarlas hasta la línea 35.

```
35 :NO_PARAM
36 echo Se debe proporcionar por parámetro el nombre del usuario
37 echo para el que hacer la copia de seguridad
38 GOTO FIN
39
40 :NO_PROFILE
41 echo No existe el directorio %UserDir%
42 echo Puede que el usuario %user% no exista.
43 net user %user%
44
45 :FIN
```

Ahí se mostrará un mensaje de error personalizado para cuando no se pasa parámetro, líneas 36 y 37, y después realiza otro salto incondicional a la etiqueta FIN que está en la línea 45. Esto lo hace para no ejecutar las líneas de la 40 a la 43, donde se da otro mensaje de error personalizado.

Si el usuario pasa un parámetro, no se cumple la condición del comando IF, no se ejecutará el comando GOTO y el script continua ejecutando la siguiente línea, la 11.

En la línea 11 se crea una variable de entorno llamada user que almacenará el valor pasado por parámetro, que es el nombre del usuario para el que queremos crear la copia de seguridad. No sabemos si este usuario existe o no, solo sabemos que se ha pasado un valor.

En la línea 12 se crea una variable de entorno llamada userDir, donde almacenará el directorio perfil de usuario del usuario, aunque para ello hace una estimación: supone que el directorio estará en el directorio [C:\Users](#) con el nombre del usuario. Esto no tiene porque ser así, pero por simplicidad lo vamos a dar por bueno.

Ahora, para evitar problemas, en la línea 14 se comprueba si el directorio almacenado en la variable userDir existe. Si no existe, dará un mensaje de error mostrando información. Para ello, en la línea 14 comprueba si existe el directorio almacenado en la variable userDir. Si no existe, entonces realizará un salto incondicional a la etiqueta NO\_PROFILE, en la línea 40.

En esta etiqueta se muestra un mensaje personalizado sobre la no existencia del directorio que suponemos como perfil del usuario y, además, se muestra información sobre el usuario para comprobar si este usuario existe.

Aquí podríamos haber utilizado otra sentencia IF con ERRORLEVEL para comprobar qué sucede al ejecutar el comando net user pasándole como parámetro el nombre del usuario para el que se quiere realizar la comprobación. Esto lo veremos en otra versión del script.

Volvemos a la línea 14. Si el directorio existe, entonces tenemos toda la información necesaria para llevar a cabo la copia de seguridad de las imágenes almacenadas en el perfil de usuario (variable userDir) que se ha pasado por parámetro (variable user).

```

10 IF "%1"==" " GOTO NO_PARAM
11 SET user=%1
12 SET userDir="C:\Users\%user%"
13
14 IF NOT EXIST %userDir% goto NO_PROFILE
15
16 REM Variables que almacenan directorio de origen, destino y fichero de inf.
17 SET dirOrigen="%userDir%"
18 SET dirDestino="C:\Backup\%user%\imagenes"
19 SET infoFile=%dirDestino%\listado.txt

```

Aquí entran en juego las variables de entorno que creamos para facilitar las modificaciones, tan solo tendremos que modificarlas con los valores almacenados con las variables creadas user y userDir.

De esta forma, dirOrigen (Línea 17) ya no almacenará el valor de %USERPROFILE%, que es la ruta del perfil del usuario actual, sino el directorio userDir, que contiene la ruta del directorio de perfil del usuario pasado por parámetro.

La variable dirDestino (18) almacena la ruta del directorio de backup del usuario cuyo nombre se pasa por parámetro, almacenado en la variable user. Ya no usamos %USERNAME%, que almacena el nombre del usuario actual.

Por último, la variable infoFile, no se tiene que cambiar, puesto que hace uso de la variable dirDestino, que ya hemos modificado.

Una vez con las variables establecidas, el script ejecuta los mismos comandos que en la versión posterior, creando el fichero de información y realizando la copia.

```

21 REM realizamos las comprobaciones previas
22 IF NOT EXIST %dirDestino% MKDIR %dirDestino%
23
24 REM Almacenamos en el fichero de información el listado de ficheros
25 REM de imágenes a copiar y la información usuario, fecha y hora.
26 SET extList=JPG BMP PNG GIF
27 FOR %%e IN (%extList%) DO DIR /S %dirOrigen%\*.*%%e >> %infoFile%
28 echo %USERNAME% %DATE% %TIME% >> %infoFile%
29
30 REM Realizamos la copia
31 FOR %%e IN (%extList%) DO XCOPY /S %dirOrigen%\*.*%%e %dirDestino%
32 echo "Copia.. OK"
33 GOTO FIN
34
35 :NO_PARAM
36 echo Se debe proporcionar por parámetro el nombre del usuario
37 echo para el que hacer la copia de seguridad
38 GOTO FIN
39
40 :NO_PROFILE
41 echo No existe el directorio %UserDir%
42 echo Puede que el usuario %user% no exista.
43 net user %user%
44
45 :FIN

```

Habría que destacar la línea 38 dónde se realiza un salto incondicional a la etiqueta FIN, que está en la línea 45. Esto es importante, puesto que si no se realiza este salto incondicional, el interprete de comandos continuará ejecutando comandos en secuencia, tras la línea 38, ejecutaría las línea 41 y 42 dónde se mostraría el mensaje de error de no parámetro. Esto sería un error.

## Comprobando el valor de vuelta de la ejecución de un comando

Con el comando IF podemos comprobar el valor de retorno de un comando. Si un comando se ejecuta correctamente, entonces su ERRORLEVEL será 0. Si no es así, su ERRORLEVEL será mayor que cero.

Con el comando IF con la variable ERRORLEVEL podemos comparar el ERRORLEVEL del último comando ejecutado con un número que se escribe a continuación. Si el ERRORLEVEL es mayor o igual que el número escrito, entonces se cumple la condición.

Como sabemos que si se ejecuta de forma correcta un comando el ERRORLEVEL es cero, entonces la comparación más simple será entre ERRORLEVEL y un 1.

```
IF ERRORLEVEL 1 echo se ha producido un error
```

Si conocemos los distintos códigos de error de una aplicación o comando que se ejecuta por línea de comandos, entonces podemos compararlos de mayor a menor.

En nuestro ejemplo de script, podemos comprobar si el usuario existe en el sistema utilizando el errorlevel.

```
SET user=%1
Net user %user% > NUL
IF ERRORLEVEL 1 GOTO NO_USER
```

Primero se ejecuta el comando net user con el nombre del usuario pasado por parámetro al script. Se utiliza una redirección de salida a NUL para que no se muestre ningún mensaje en pantalla. NUL es un fichero especial, que no se guarda. Es como un fichero en el que lo que se escriba en él, se pierde. No se almacena en ninguna parte.

Si el comando anterior produce un error, entonces la variable ERRORLEVEL almacenará un valor mayor de 0. No conocemos el código exacto de ERRORLEVEL para un error en el que no exista el usuario que se está consultando, no obstante sabemos que el error será mayor que 0. Podríamos mostrar el ERRORLEVEL por curiosidad en una consola ejecutando el comando y mostrando con ECHO el valor de ERRORLEVEL

```
NET USER NO_USUARIO
ECHO %ERRORLEVEL%
```

Cuidado, porque esto no podemos hacerlo en un script, puesto que perderíamos el valor de ERRORLEVEL del comando NET USER. Una vez ejecutado el comando ECHO, el valor de ERRORLEVEL cambiará al valor devuelto por el último comando ejecutado, es decir ECHO.

Volvemos a la comprobación de ERRORLEVEL, si el comando anterior, NET USER, da error el comando IF ERRORLEVEL 1 cumplirá la condición y se ejecutará el comando escrito a continuación. Lo ideal es utilizar un salto incondicional a la etiqueta NO\_USER, donde se mostraría un mensaje de error personalizado.

```
10 IF "%1"==" " GOTO NO_PARAM
11 SET user=%1
12 NET USER %user% > NUL
13 IF ERRORLEVEL 1 GOTO NO_USER
```

Mensaje de error personalizado:

```
42 :NO_USER
43 echo El usuario %user% no tiene cuenta en el sistema
44 echo Se debe elegir un usuario con cuenta en el sistema
45 GOTO FIN
```

En la siguiente figura tenemos el script con esta nueva modificación. Se podrían mejorar algunos aspectos, pero esto lo dejamos a consideración de los lectores :).

```
7 REM Autor: Javier Pedrero Martín
8 REM Fecha: 29/01/2019
9 REM Comprobamos si se pasa por parámetro nombre de usuario y si existe su perfil
10 IF "%1"==" " GOTO NO_PARAM
11 SET user=%1
12 NET USER %user% > NUL
13 IF ERRORLEVEL 1 GOTO NO_USER
14
15 SET userDir="C:\Users\%user%"
16 IF NOT EXIST %userDir% goto NO_PROFILE
17
18 REM Variables que almacenan directorio de origen, destino y fichero de inf.
19 SET dirOrigen="%userDir%"
20 SET dirDestino="C:\Backup\%user%\imagenes"
21 SET infoFile=%dirDestino%\listado.txt
22
23 REM realizamos las comprobaciones previas
24 IF NOT EXIST %dirDestino% MKDIR %dirDestino%
25
26 REM Almacenamos en el fichero de información el listado de ficheros
27 REM de imágenes a copiar y la información usuario, fecha y hora.
28 SET extList=JPG BMP PNG GIF
29 FOR %%e IN (%extList%) DO DIR /S %dirOrigen%\*.*%%e >> %infoFile%
30 echo %USERNAME% %DATE% %TIME% >> %infoFile%
31
32 REM Realizamos la copia
33 FOR %%e IN (%extList%) DO XCOPY /S %dirOrigen%\*.*%%e %dirDestino%
34 echo "Copia.. OK"
35 GOTO FIN
36
37 :NO_PARAM
38 echo Se debe proporcionar por parámetro el nombre del usuario
39 echo para el que hacer la copia de seguridad
40 GOTO FIN
41
42 :NO_USER
43 echo El usuario %user% no tiene cuenta en el sistema
44 echo Se debe elegir un usuario con cuenta en el sistema
45 GOTO FIN
46
47 :NO_PROFILE
48 echo No existe el directorio %UserDir%
49 echo No se puede realizar la copia. Contacte con el administrador.
50
51 :FIN
```

## ¿Cómo comprobar que el usuario tiene privilegios de administrador?

Esto es interesante, puesto que la nueva versión del script debería ser ejecutada por un usuario que tenga permisos para acceder a los perfiles de los usuarios, es decir un administrador.

Para comprobar si somos administrador la forma más sencilla sería consultar algún comando que nos diga si somos administrador. No obstante, si tratamos de consultar variables de entorno como USERNAME nos dirá que nuestro usuario es el usuario con el que trabajamos habitualmente, no administrador.

La opción aquí es tratar de realizar alguna operación que necesite permisos de administrador, que no produzca efectos secundarios y comprobar si se ha podido realizar o no con IF ERRORLEVEL. Si no se pudo realizar no tenemos privilegios de administrador.

En el siguiente [enlace de stackoverflow](#), se proporciona una solución para comprobar si tenemos permisos de administrador e incluso elevar nuestros permisos. Es una solución elaborada y, quizás se escape de los objetivos de este documento, pero es interesante si algún lector quiere profundizar un poco en el uso de ciertos comandos y soluciones.

La solución, resumida, sería comprobar si se puede ejecutar el comando NET FILE, redirigiendo la salida de información y la de errores a NUL. Este comando requiere permisos de administrador, así que si se da un ERRORLEVEL distinto de 0 no tenemos privilegios de administrador.

```
NET FILE 1>NUL 2>NUL
IF ERRORLEVEL 1 GOTO NO_ADMIN
```

Podríamos haberlo simplificado con

```
NET FILE > NUL
```

Pero puede que se muestre algún mensaje incómodo en pantalla. El número 1 y 2 delante de las redirecciones indica que se redireccionará la salida estándar (1) y la salida de errores (2) a un fichero, en ambos casos el fichero NUL, que ya hemos explicado.

Es importante utilizar las dos redirecciones puesto que, si tenemos permisos de administrador, se mostrará un mensaje por la salida estándar. Si ejecutamos con usuario sin privilegios, se mostrará un mensaje de error por la salida de errores.

La solución simple consistiría en realizar la comprobación al inicio del script, antes de realizar ninguna otra comprobación o acción.

```
10 NET FILE 1>NUL 2>NUL
11 IF ERRORLEVEL 1 GOTO NO_ADMIN
```

Si se produce error, entonces salta incondicionalmente a la etiqueta NO\_ADMIN, donde se mostrará el mensaje de error correspondiente y saltará a la etiqueta final para no mostrar más mensajes de error.

```
40 :NO_ADMIN
41 echo Se debe ejecutar este script con privilegios de administrador
42 goto FIN
```

```
10 NET FILE 1>NUL 2>NUL
11 IF ERRORLEVEL 1 GOTO NO_ADMIN
12
13 IF "%1"==" " GOTO NO_PARAM
14 SET user=%1
15 NET USER %user% > NUL
16 IF ERRORLEVEL 1 GOTO NO_USER
17
18 SET userDir="C:\Users\%user%"
19 IF NOT EXIST %userDir% goto NO_PROFILE
20
21 REM Variables que almacenan directorio de origen, destino y fichero de inf.
22 SET dirOrigen="%userDir%"
23 SET dirDestino="C:\Backup\%user%\imagenes"
24 SET infoFile=%dirDestino%\listado.txt
25
26 REM realizamos las comprobaciones previas
27 IF NOT EXIST %dirDestino% MKDIR %dirDestino%
28
29 REM Almacenamos en el fichero de información el listado de ficheros
30 REM de imágenes a copiar y la información usuario, fecha y hora.
31 SET extList=JPG BMP PNG GIF
32 FOR %%e IN (%extList%) DO DIR /S %dirOrigen%\*.*%%e >> %infoFile%
33 echo %USERNAME% %DATE% %TIME% >> %infoFile%
34
35 REM Realizamos la copia
36 FOR %%e IN (%extList%) DO XCOPY /S %dirOrigen%\*.*%%e %dirDestino%
37 echo "Copia.. OK"
38 GOTO FIN
39
40 :NO_ADMIN
41 echo Se debe ejecutar este script con privilegios de administrador
42 goto FIN
43
44 :NO_PARAM
45 echo Se debe proporcionar por parámetro el nombre del usuario
46 echo para el que hacer la copia de seguridad
47 GOTO FIN
48
49 :NO_USER
50 echo El usuario %user% no tiene cuenta en el sistema
51 echo Se debe elegir un usuario con cuenta en el sistema
52 GOTO FIN
53
54 :NO_PROFILE
55 echo No existe el directorio %UserDir%
56 echo No se puede realizar la copia. Contacte con el administrador.
57
58 :FIN
```