POWERSHELL

ALEJANDRO SAINZ SAINZ

Contenido

[1. APARTADO 1 2](#_Toc185490052)

[1.1 CREANDO LA ESTRUCTURA DE DIRECTORIOS 2](#_Toc185490053)

[2 APARTADO 2 5](#_Toc185490054)

[2.1 DESCARGANDO IMÁGENES 5](#_Toc185490055)

[3 APARTADO 3 6](#_Toc185490056)

[3.1 DESCARGANDO DOCUMENTOS 6](#_Toc185490057)

[4 APARTADO 4 7](#_Toc185490058)

[4.1 MOVIENDO DOCUMENTOS 7](#_Toc185490059)

[5 APARTADO 5 9](#_Toc185490060)

[5.1 FILTRANDO ARCHIVOS 9](#_Toc185490061)

[6 APARTADO 6 10](#_Toc185490062)

[6.1 FILTRANDO PDF… 10](#_Toc185490063)

[7 APARTADO 7 11](#_Toc185490064)

[7.1 FILTRANDO IMÁGENES POR TAMAÑO 11](#_Toc185490065)

[8 APARTADO 8 13](#_Toc185490066)

[8.1 DOCUMENTO MOVIL 13](#_Toc185490067)

[9 APARTADO 9 15](#_Toc185490068)

[9.1 MOVERTAREACOMPLETADA 15](#_Toc185490069)

[10 APARTADO 10 16](#_Toc185490070)

[10.1 DOCUMENTACIÓN 16](#_Toc185490071)

[11 APARTADO 11 16](#_Toc185490072)

[11.1 FINALZIP 16](#_Toc185490073)

# APARTADO 1

## CREANDO LA ESTRUCTURA DE DIRECTORIOS

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Poco a poco voy creando la estructura de directorios.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Unos cuantos más y listo.

# APARTADO 2

## DESCARGANDO IMÁGENES

2. Descarga 5 imágenes pesadas y 5 imágenes de tamaño normal desde google. (esta acción NO se realizará desde PowerShell)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Estas 10 imágenes he descargado.

# APARTADO 3

## DESCARGANDO DOCUMENTOS

3. Descarga dos documentos Word y dos documentos PDF desde Google Scholar. (esta

acción NO se realizará desde PowerShell)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Descargados

# APARTADO 4

## MOVIENDO DOCUMENTOS

4. Mueve todos los archivos descargados desde la carpeta de descargas del dispositivo a la carpeta Documentos en la estructura creada.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

En principio con ese comando debería de funcionar. Uso un dir para comprobar.

Texto

Descripción generada automáticamente

Ahí lo tenemos todo usando dir .. Veo que me equivoque y tengo que moverlos a documentos

Texto

Descripción generada automáticamente con confianza media

Lo arreglo

Texto

Descripción generada automáticamenteListo.

# APARTADO 5

## FILTRANDO ARCHIVOS

5. Filtra los archivos en Documentos por tipo de archivo Word y muévelos a la carpeta

DOCS.

Para esto tendré que usar el comando move pero indicando que son todos los archivos \*.docx.

Texto

Descripción generada automáticamente

Uso este comando

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Uso dir y compruebo

# APARTADO 6

## FILTRANDO PDF…

6. Filtra los archivos por tipo de archivo PDF y muévelos a la carpeta PDFS.

Mismo caso que en el ejercicio anterior pero con \*.pdf

Texto

Descripción generada automáticamente

Compruebo con dir

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

# APARTADO 7

## FILTRANDO IMÁGENES POR TAMAÑO

7. Filtra las imágenes restantes en Documentos por tamaño. Mueve aquellas que superen

1 MB a la carpeta Objetos Pesados.

En principio, para este ejercicio tengo que usar las denominadas pipelines, o lo que seria la concatenación de comandos. Vamos a verlo poco a poco:

Para los ejercicios anteriores había usado los comandos propios de CMD, que también funcionan en PowerShell, pero hay algunos nuevos, que permiten hacer las cosas de una mejor forma.

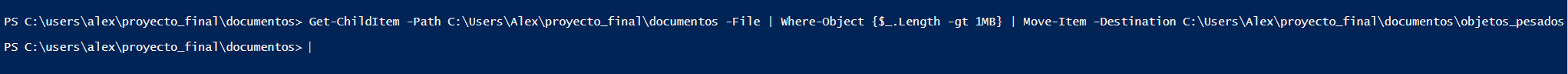
Texto

Descripción generada automáticamente

Para empezar a entender el comando poco a poco vamos a desmenuzarlo, que yo tampoco lo conocía y todavía tengo que entenderlo bien.

1. Get-ChildItem -File: Con este comando lo que hacemos es seleccionar todos los elementos que queramos y con -file indicamos que tienen que ser ficheros y no directorios
2. | : Símbolo de la pipeline, quiere decir que al resultado del comando que le precede, hay que aplicarle el efecto del comando que vienen a continuación.
3. Where-Object: Con esto indicamos que los objetos van a necesitar cumplir una condición.
4. $\_.Length: filtra los archivos obtenidos por el tamaño de los mismos.
5. -gt: Greater than o mayor que, para que los archivos cumplan que su tamaño sea mayor que
6. 1MB: el tamaño que deben de superar los archivos que hemos filtrado.

Una vez hecho esto, que yo no conocía hasta hoy, hay que seguir con la siguiente parte del enunciado, moverlos a la carpeta objetos pesados.



Bueno, el comando es muy largo, así que voy a escribir aquí:

Después de introducir el comando anterior usaría lo siguientes:



Al filtrar los objetos de la búsqueda, con el símbolo | encadenaría un segundo comando, Move-Item.

Con este comando seguido de -Destination [Ruta Destino] le indico a donde quiero mover los objetos resultantes de ejecutar la primera parte del comando.

Ahora debería de hacer un dir para comprobarlo.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Aquí vemos que el resultado ha sido el correcto.

# APARTADO 8

## DOCUMENTO MOVIL

8. Crea un documento de texto en la primera de las carpetas (Tareas Por Hacer). Hazlo

moverse por cada una de las carpetas pertenecientes a “Tareas Pendientes” y en cada

de las dos carpetas que servirá como destino deberás añadirle una línea de texto al

archivo.

Lo primero, en este ejercicio, es aprender a crear variables, almacenar datos en ellas y luego a mostrarlo por pantalla.

Texto

Descripción generada automáticamente

El primer comando de la imagen es declarar una variable con $[nombre\_variable], en este caso nombreDirectorio.

Después le indico que esta variable es igual al resultado de obtener los nombres de subdirectorios dentro de la carpeta que le he indicado según el argumento -Path.

Con el comando write-output ordeno que se muestre lo que está almacenado en la variable $nombreDirectorio.

Ahora ya solo va quedando aprender a hacer bucles en la PowerShell.

Pero antes de eso busco como ordenarlos de forma descendente para que la última carpeta sea tareas completadas. Esos se consigue usando al final del anterior comando | Sort-object -Descending

Texto

Descripción generada automáticamente

En la captura anterior se puede ver el resultado.

Ahora si, hay que aprender sobre los bucles, que serán parecidos a todos los lenguajes.

Texto

Descripción generada automáticamente

Esta es la estructura de un for. Lo que le he pedido es que cada vez que se produce una iteración Muestre el texto Estamos en y luego el nombre de la carpeta, que ya estaban ordenadas de la forma que yo quería.

Ahora que he probado eso, voy a tener que introducir dentro del bucle, que mueva el archivo que voy a crear y que añada una línea a ese archivo cada vez que vaya a una carpeta.

Además, tengo que asegurarme de que el archivo esté en Tareas por Hacer, así que debo excluirlo de la lista que tengo almacenada en la variable.

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

Creo el archivo en tareas por hacer.



Redefino lo que se almacena en la variable.

Texto

Descripción generada automáticamente

Ya la tengo como quiero. Por ahora.

No consigo de ninguna forma hacer el for.

# APARTADO 9

## MOVERTAREACOMPLETADA

9. Crea una función personalizada llamada MoverTareaCompletada que se encargue de

mover automáticamente los archivos desde la carpeta Tareas Completadas hacia la

carpeta Entregable Proyecto.

Get-ChildItem C:\Users\daw1.COMPU-S026\proyecto\_final\tareas\_pendientes\tareas\_completadas\\*.\* | Move-Item -Destination C:\Users\daw1.COMPU-S026\proyecto\_final\entregable\_proyecto

Con este comando tan corto, creándolo dentro del script, se supone que seleccione todos los archivos de la carpeta de origen y lo mueva a la carpeta elegida como destino.

# APARTADO 10

## DOCUMENTACIÓN

10. Documenta cada paso del proceso en un archivo de texto llamado InformeProyecto.docx dentro de la carpeta Entregable Proyecto. Incluye explicaciones claras, capturas de

pantalla de los resultados y comentarios sobre desafíos y lecciones aprendidas durante

la práctica.

Supongo que este documento es ese mismo documento.

# APARTADO 11

## FINALZIP

11. Crea una función personalizada llamada FINALZIP que comprima la carpeta Proyecto

Final en un archivo ZIP llamado EntregableProyecto.zip.

Compress-Archive -Path C:\Users\daw1.COMPU-S026\proyecto\_final\entregable\_proyecto -DestinationPath C:\Users\daw1.COMPU-S026\proyecto\_final\EntregableProyecto.zip

Con otro comando, también muy corto, podemos crear el archivo comprimido.