



POWERSHELL

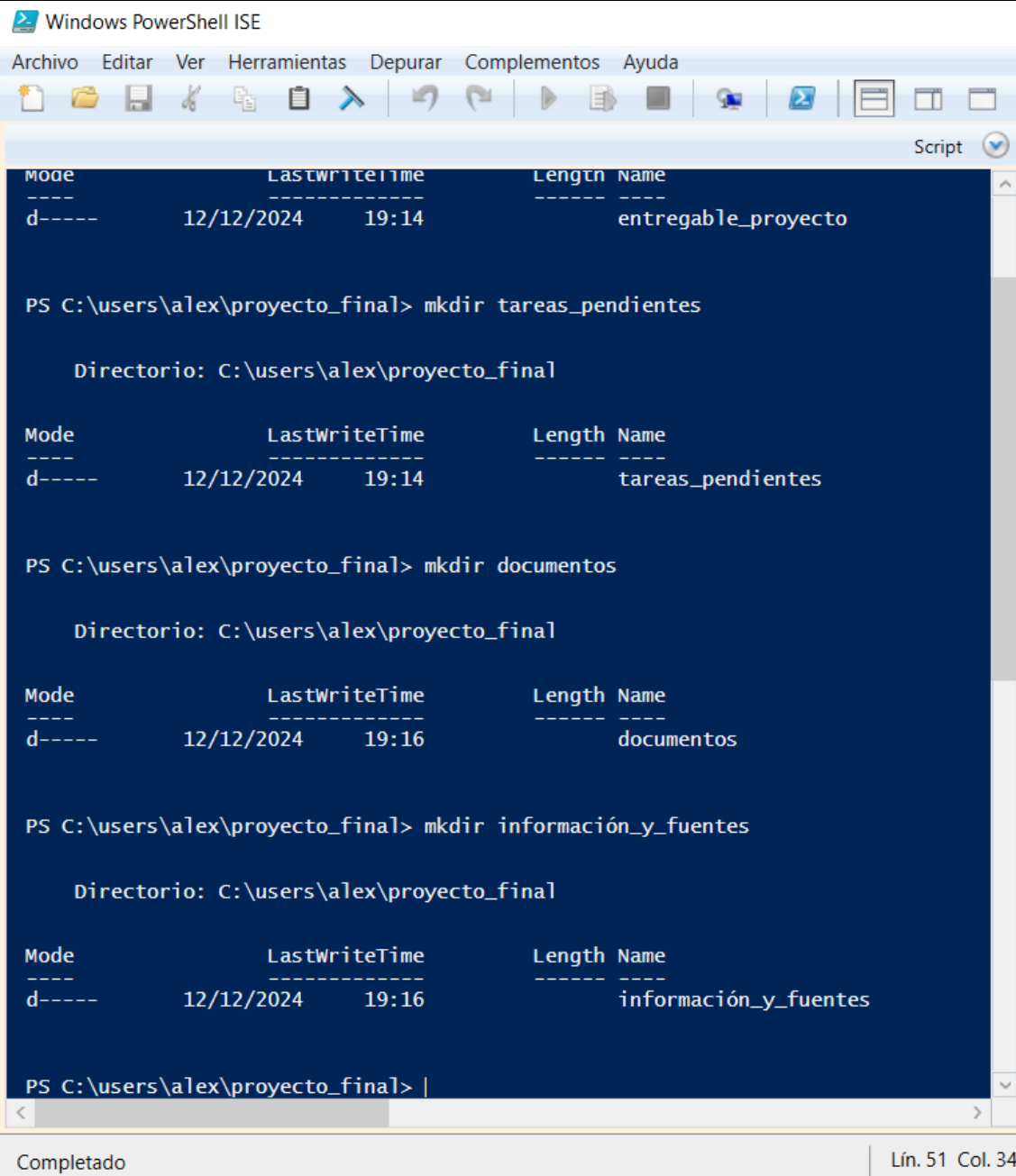
ALEJANDRO SAINZ SAINZ

1 CONTENIDO

1.	APARTADO 1	2
1.1	CREANDO LA ESTRUCTURA DE DIRECTORIOS	2
2	APARTADO 2	5
2.1	DESCARGANDO IMÁGENES	5
3	APARTADO 3	6
3.1	DESCARGANDO DOCUMENTOS	6
4	APARTADO 4	7
4.1	MOVIENDO DOCUMENTOS	7
5	APARTADO 5	9
5.1	FILTRANDO ARCHIVOS	9
6	APARTADO 6	10
6.1	FILTRANDO PDF... ..	10
7	APARTADO 7	11
7.1	FILTRANDO IMÁGENES POR TAMAÑO	11
8	APARTADO 8	13
8.1	DOCUMENTO MOVIL	13
9	APARTADO 9	15
9.1	MOVERTAREACOMPLETADA	15
10	APARTADO 10	16
10.1	DOCUMENTACIÓN	16
11	APARTADO 11	16
11.1	FINALZIP	16

1. APARTADO 1

1.1 CREANDO LA ESTRUCTURA DE DIRECTORIOS



The screenshot shows the Windows PowerShell ISE interface. The menu bar includes 'Archivo', 'Editar', 'Ver', 'Herramientas', 'Depurar', 'Complementos', and 'Ayuda'. The toolbar contains icons for file operations and execution. The main console area has a dark blue background and displays the following commands and output:

```
Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024    19:14         entregable_proyecto

PS C:\users\alex\proyecto_final> mkdir tareas_pendientes

Directorio: C:\users\alex\proyecto_final

Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024    19:14         tareas_pendientes

PS C:\users\alex\proyecto_final> mkdir documentos

Directorio: C:\users\alex\proyecto_final

Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024    19:16         documentos

PS C:\users\alex\proyecto_final> mkdir información_y_fuentes

Directorio: C:\users\alex\proyecto_final

Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024    19:16         información_y_fuentes

PS C:\users\alex\proyecto_final> |
```

At the bottom of the console, there is a status bar with the text 'Completado' on the left and 'Lín. 51 Col. 34' on the right.

Poco a poco voy creando la estructura de directorios.

```
PS C:\users\alex\proyecto_final> dir

    Directorio: C:\users\alex\proyecto_final

Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024    19:18      documentos
d-----         12/12/2024    19:14  entregable_proyecto
d-----         12/12/2024    19:16  información_y_fuentes
d-----         12/12/2024    19:18      tareas_pendientes

PS C:\users\alex\proyecto_final> cd información_y_fuentes
PS C:\users\alex\proyecto_final\información_y_fuentes> mkdir DOCS

    Directorio: C:\users\alex\proyecto_final\información_y_fuentes

Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024    19:20      DOCS

PS C:\users\alex\proyecto_final\información_y_fuentes> mkdir PDFs

    Directorio: C:\users\alex\proyecto_final\información_y_fuentes

Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024    19:20      PDFs

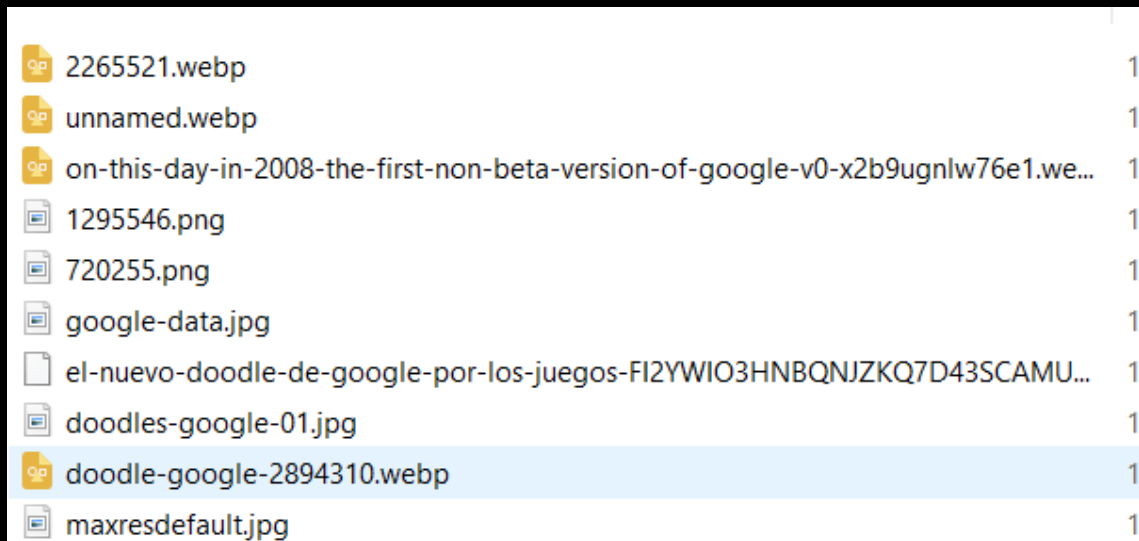
PS C:\users\alex\proyecto_final\información_y_fuentes> |
```

Unos cuantos más y listo.

2 APARTADO 2

2.1 DESCARGANDO IMÁGENES

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

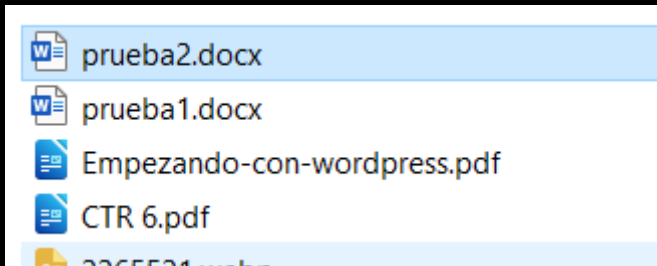


Estas 10 imágenes he descargado.

3 APARTADO 3

3.1 DESCARGANDO DOCUMENTOS

3. Descarga dos documentos Word y dos documentos PDF desde Google Scholar. (esta acción NO se realizará desde PowerShell)



Descargados

4 APARTADO 4

4.1 MOVIENDO DOCUMENTOS

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

```
PS C:\users\alex\proyecto_final\información_y_fuentes> copy C:\Users\Alex\Downloads\*.* C:\Users\Alex\proyecto_final
PS C:\users\alex\proyecto_final\información_y_fuentes> |
```

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

```
PS C:\users\alex\proyecto_final\información_y_fuentes> dir ..

Directorio: C:\users\alex\proyecto_final

Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024        19:18             documentos
d-----         12/12/2024        19:14             entregable_proyecto
d-----         12/12/2024        19:20             información_y_fuentes
d-----         12/12/2024        19:18             tareas_pendientes
-a-----         12/12/2024        19:23           15233 1295546.png
-a-----         12/12/2024        19:23           5254 2265521.webp
-a-----         12/12/2024        19:23           15322 720255.png
-a-----         12/12/2024        19:28          186237 CTR 6.pdf
-a-----         12/12/2024        19:22          577754 doodle-google-2894310.webp
-a-----         12/12/2024        19:22          177712 doodles-google-01.jpg
-a-----         12/12/2024        19:22          25170 el-nuevo-doodle-de-google-por-los-juegos-FI2YWI03HNBQNJZKQ7D43SCAMU.avif
-a-----         12/12/2024        19:29          2528378 Empezando-con-wordpress.pdf
-a-----         12/12/2024        19:22          1131192 google-data.jpg
-a-----         12/12/2024        19:22          140285 maxresdefault.jpg
-a-----         12/12/2024        19:23          10310 on-this-day-in-2008-the-first-non-beta-version-of-google-v0-x2b9ugn1w76e1.w
-a-----         12/12/2024        19:31              0 prueba1.docx
-a-----         12/12/2024        19:31              0 prueba2.docx
-a-----         12/12/2024        19:23           4688 unnamed.webp

PS C:\users\alex\proyecto_final\información_y_fuentes> |
```

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

```
PS C:\users\alex\proyecto_final\información_y_fuentes> cd..
PS C:\users\alex\proyecto_final> copy *.* .\documentos
PS C:\users\alex\proyecto_final> |
```


Lo arreglo

```
PS C:\users\alex\proyecto_final> dir .\documentos

Directorio: C:\users\alex\proyecto_final\documentos

Mode                LastWriteTime         Length Name
----                -
d-----          12/12/2024      19:18             objetos_pesados
-a-----          12/12/2024      19:23           15233 1295546.png
-a-----          12/12/2024      19:23           5254 2265521.webp
-a-----          12/12/2024      19:23           15322 720255.png
-a-----          12/12/2024      19:28          186237 CTR 6.pdf
-a-----          12/12/2024      19:22          577754 doodle-google-2894310.webp
-a-----          12/12/2024      19:22          177712 doodles-google-01.jpg
-a-----          12/12/2024      19:22           25170 el-nuevo-doodle-de-google-por-los-juegos-FI2YWIO3HNBQNJZKQ7D43SCAMU.avif
-a-----          12/12/2024      19:29          2528378 Empezando-con-wordpress.pdf
-a-----          12/12/2024      19:22          1131192 google-data.jpg
-a-----          12/12/2024      19:22          140285 maxresdefault.jpg
-a-----          12/12/2024      19:23           10310 on-this-day-in-2008-the-first-non-beta-version-of-google-v0-x2b9ugn1w76e1.webp
-a-----          12/12/2024      19:31            0 prueba1.docx
-a-----          12/12/2024      19:31            0 prueba2.docx
-a-----          12/12/2024      19:23           4688 unnamed.webp
```

Listo.

5 APARTADO 5

5.1 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.

```
PS C:\users\alex\proyecto_final> cd documentos
PS C:\users\alex\proyecto_final\documentos> copy *.docx ..\información_y_fuentes\DOCS\
```

Uso este comando

```
PS C:\users\alex\proyecto_final\documentos> copy *.docx ..\información_y_fuentes\DOCS\
PS C:\users\alex\proyecto_final\documentos> dir ..\información_y_fuentes\DOCS

Directorio: C:\users\alex\proyecto_final\información_y_fuentes\DOCS

Mode                LastWriteTime         Length Name
----                -
-a-----         12/12/2024   19:31             0 prueba1.docx
-a-----         12/12/2024   19:31             0 prueba2.docx

PS C:\users\alex\proyecto_final\documentos> |
```

Uso dir y compruebo

6 APARTADO 6

6.1 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

```
PS C:\users\alex\proyecto_final\documentos> copy *.pdf ..\información_y_fuentes\PDFs
```

Compruebo con dir

```
PS C:\users\alex\proyecto_final\documentos> dir ..\información_y_fuentes\PDFs

Directorio: C:\users\alex\proyecto_final\información_y_fuentes\PDFs

Mode                LastWriteTime         Length Name
----                -
-a----           12/12/2024   19:28         186237 CTR 6.pdf
-a----           12/12/2024   19:29        2528378 Empezando-con-wordpress.pdf

PS C:\users\alex\proyecto_final\documentos> |
```

7 APARTADO 7

7.1 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 sería 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.

```
PS C:\users\alex\proyecto_final\documentos> Get-ChildItem -file | Where-Object {$_.Length -gt 1MB}

Directorio: C:\users\alex\proyecto_final\documentos

Mode                LastWriteTime         Length Name
----                -
-a----          12/12/2024   19:29       2528378 Empezando-con-wordpress.pdf
-a----          12/12/2024   19:22       1131192 google-data.jpg

PS C:\users\alex\proyecto_final\documentos>
```

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. `$_`: 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.

```
PS C:\users\alex\proyecto_final\documentos> Get-ChildItem -Path C:\Users\Alex\proyecto_final\documentos -File | Where-Object {$_.Length -gt 1MB} | Move-Item -Destination C:\Users\Alex\proyecto_final\documentos\objetos_pesados
PS C:\users\alex\proyecto_final\documentos> |
```

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

Después de introducir el comando anterior usaría lo siguientes:

```
| Move-Item -Destination C:\Users\Alex\proyecto_final\documentos\objetos_pesados
```

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.

```
Directorio: C:\users\alex\proyecto_final\documentos\objetos_pesados

Mode                LastWriteTime         Length Name
----                -
-a-----         12/12/2024   19:29         2528378 Empezando-con-wordpress.pdf
-a-----         12/12/2024   19:22         1131192 google-data.jpg

PS C:\users\alex\proyecto_final\documentos> |
```

Aquí vemos que el resultado ha sido el correcto.

8 APARTADO 8

8.1 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.

```
PS C:\users\alex\proyecto_final\tareas_pendientes> $nombreDirectorio = Get-ChildItem -Path C:\Users\Alex\proyecto_final\tareas_pendientes
PS C:\users\alex\proyecto_final\tareas_pendientes> Write-Output $nombreDirectorio

Directorio: C:\Users\Alex\proyecto_final\tareas_pendientes
```

Mode	LastWriteTime	Length	Name
d----	12/12/2024 19:17		tareas_completadas
d----	12/12/2024 19:17		tareas_en_curso
d----	12/12/2024 19:18		tareas_por_hacer

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

```
PS C:\users\alex\proyecto_final\tareas_pendientes> $nombreDirectorio = Get-ChildItem -Path C:\Users\Alex\proyecto_final\tareas_pendientes | Sort-object -Descending
PS C:\users\alex\proyecto_final\tareas_pendientes> Write-Output $nombreDirectorio

Directorio: C:\Users\Alex\proyecto_final\tareas_pendientes
```

Mode	LastWriteTime	Length	Name
d----	12/12/2024 19:18		tareas_por_hacer
d----	12/12/2024 19:17		tareas_en_curso
d----	12/12/2024 19:17		tareas_completadas

```
PS C:\users\alex\proyecto_final\tareas_pendientes> |
```

En la captura anterior se puede ver el resultado.

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

```
PS C:\users\alex\proyecto_final\tareas_pendientes> for($i = 0; $i -le $nombreDirectorio.Length; $i++){echo Estamos en $nombreDirectorio[$i]}
Estamos
en

    Directorio: C:\Users\Alex\proyecto_final\tareas_pendientes

Mode                LastWriteTime         Length Name
----                -
d-----          12/12/2024    19:18             tareas_por_hacer
Estamos
en
d-----          12/12/2024    19:17             tareas_en_curso
Estamos
en
d-----          12/12/2024    19:17             tareas_completadas
Estamos
en

PS C:\users\alex\proyecto_final\tareas_pendientes> |
```

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.

```
PS C:\Users\Alex\proyecto_final\tareas_pendientes\tareas_por_hacer> ni prueba.txt

    Directorio: C:\Users\Alex\proyecto_final\tareas_pendientes\tareas_por_hacer

Mode                LastWriteTime         Length Name
----                -
-a-----          16/12/2024    19:51             0 prueba.txt

PS C:\Users\Alex\proyecto_final\tareas_pendientes\tareas_por_hacer> |
```

Creo el archivo en tareas por hacer.

```
$nombreDirectorio = Get-ChildItem -Path C:\Users\Alex\proyecto_final\tareas_pendientes -Exclude C:\Users\Alex\proyecto_final\tareas_pendientes\tareas_por_hacer | Sort-Object -Descending
```

Redefino lo que se almacena en la variable.

```
Directorio: C:\Users\Alex\proyecto_final\tareas_pendientes

Mode                LastWriteTime         Length Name
----                -
d-----         12/12/2024         19:17      tareas_en_curso
d-----         12/12/2024         19:17      tareas_completadas

PS C:\users\alex\proyecto_final\tareas_pendientes> |
```

Ya la tengo como quiero. Por ahora.

No consigo de ninguna forma hacer el for.

9 APARTADO 9

9.1 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.

10 APARTADO 10

10.1 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.

11 APARTADO 11

11.1 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.