|  |  |  |
| --- | --- | --- |
| IES San Andrés (@iessanandres) | Twitter | **Ciclo**: | Administración de Sistemas Informáticos en Red |
| **Módulo**: | IMPLANTACIÓN DE SISTEMAS OPERATIVOS |
| **Grupo**: | 1º ASIR |
| **PR0302: Introducción a Powershell (II)** | | |

|  |  |
| --- | --- |
| **Nombre/s:** | Adrián Aparicio Vivas |

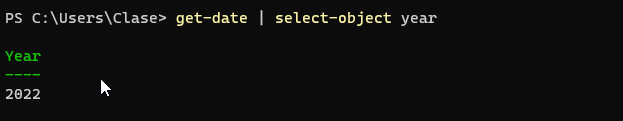
Tienes que contestar las preguntas en este mismo fichero después de cada pregunta. No te olvides de poner tu nombre en el recuadro superior.

Cuando hayas acabado todas las prácticas renombras el fichero para que se llame **{Apellido1} {Apellido2}, {Nombre} – PR0302**. En el nombre y apellidos la primera mayúscula y el resto en minúsculas. El fichero tiene que estar en formato PDF. **Cualquier fichero que no siga esta nomenclatura o no esté en PDF no será corregido**. El fichero final lo tienes que subir a la plataforma.

|  |
| --- |
| **Ejercicio 1: Powershell** |

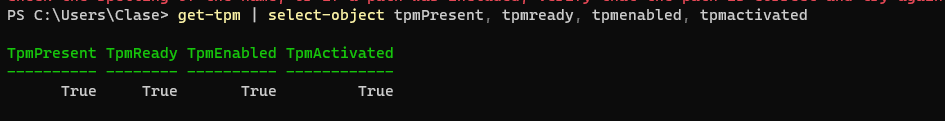
Realiza las siguientes tareas que se te piden utilizando Powershell. Para contestar lo mejor es que hagas una captura de pantalla donde se vea el comando que has introducido y las primeras líneas de la salida de este.

**1.-** El comando **Get-Date** muestra la fecha y hora actual. Muestra por pantalla únicamente el año en que estamos.



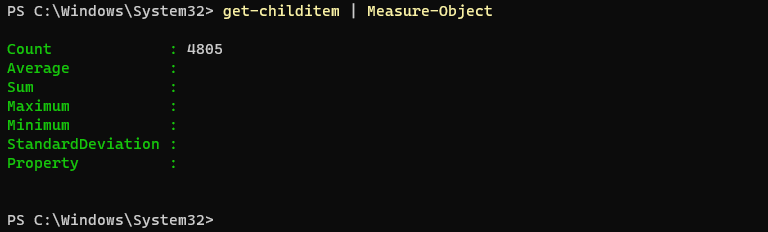
**2.-** Uno de los requisitos de Windows 11 es que es procesador tenga TPM habilitado. Powershell dispone del comando **Get-TPM** que nos muestra información sobre este módulo. Muestra por pantalla, en formato tabla, las propiedades **TpmPresent**, **TpmReady**, **TpmEnabled** y **TpmActivated**.

- Antes de nada hay que abrir la terminal en modo administrador.

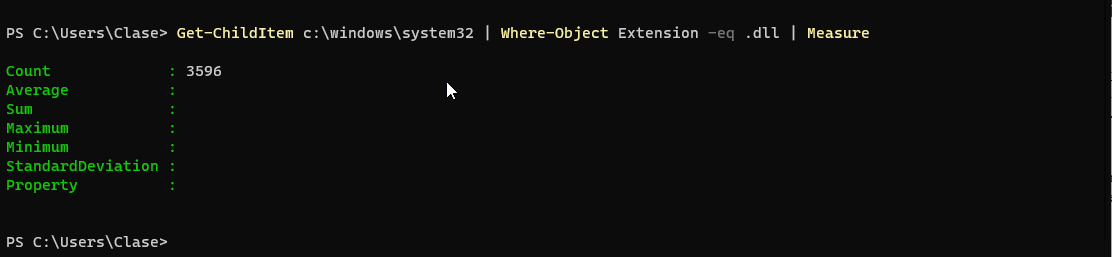


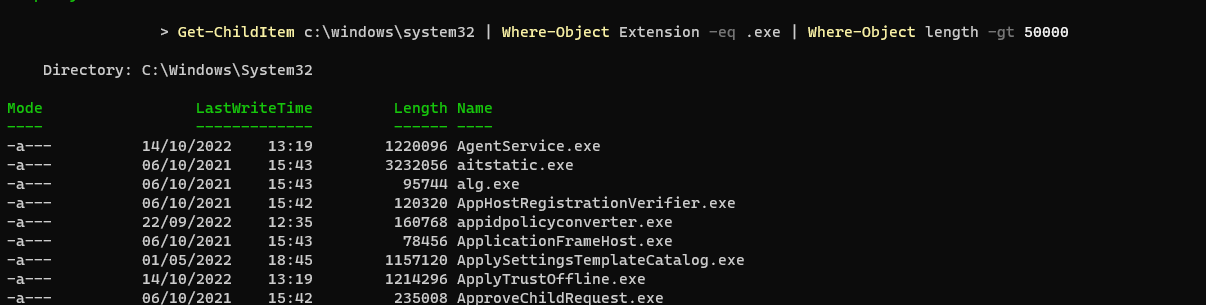
En los siguientes ejercicios trabajaremos con los ficheros devueltos por el comando **Get-ChildItem C:\Windows\System32**.

**3.-** Muestra por pantalla el número de ficheros y directorios que hay en ese directorio.

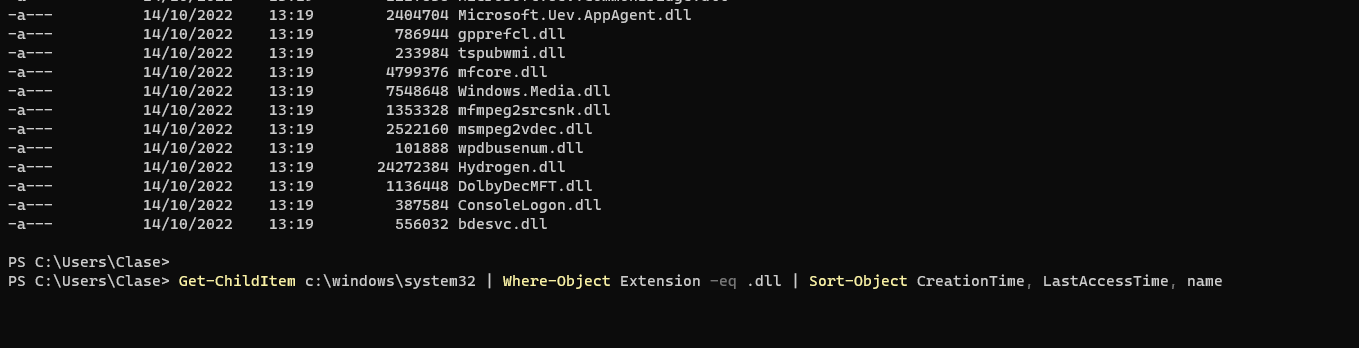


**4.-** Los objetos devueltos por el comando anterior tienen una propiedad denominada **Extension**, que indica la extensión del archivo. Calcula el número de ficheros en el directorio que tienen la extensión **.dll**.

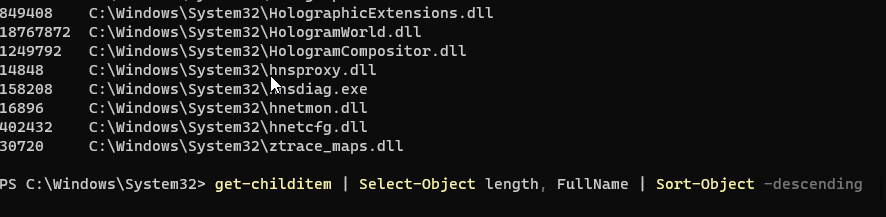
**5.-** Muestra los ficheros del directorio con extensión **.exe** que tengan un tamaño superior a 50000 bytes.



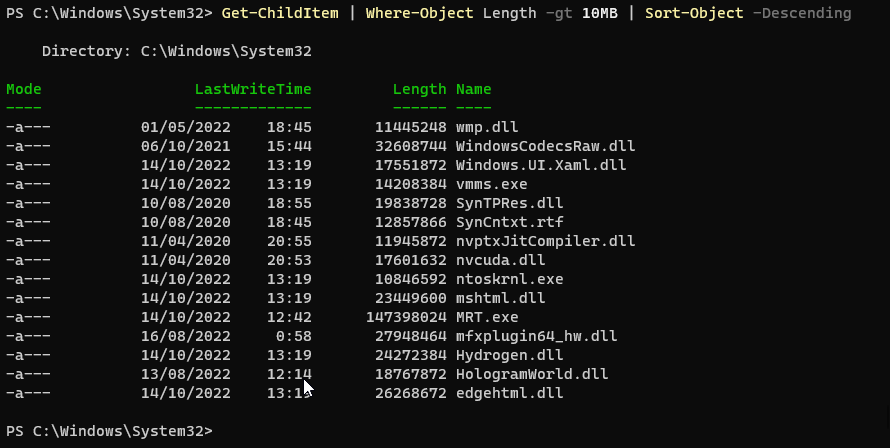
**6.-** Muestra los ficheros de este directorio que tengan extensión **.dll**, ordenados por fecha de creación y mostrando únicamente las propiedades de fecha de creación (*CreationTime*), último acceso (*LastAccessTime*) y nombre (*Name*).



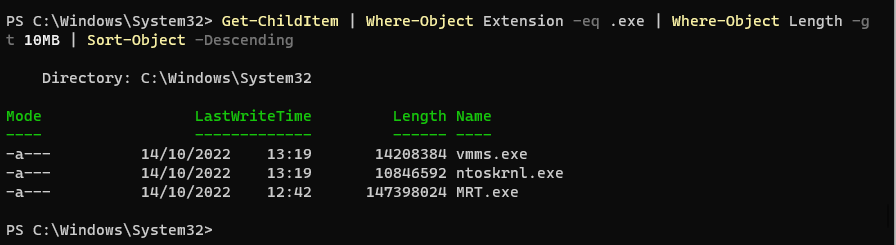
**7.-** Muestra el tamaño (*Length*) y nombre completo (*FullName*) de todos los ficheros del directorio ordenados por tamaño en sentido descendente.



**8.**- Muestra el tamaño y nombre completo de todos los ficheros del directorio que tengan un tamaño superior a 10MB (10000000 bytes) ordenados por tamaño.



**9.-** Muestra el tamaño y nombre completo de todos los ficheros del directorio que tengan un tamaño superior a 10MB y extensión .exe ordenados por tamaño.



Hemos visto cómo usar el comando Where-Object para filtrar objetos con propiedades de tipo texto o numérico (por ejemplo, Where-Object CPU -gt 1 o Where-Object Name -eq “Notepad”, sin embargo, hay propiedades que pueden tener otro tipo de datos. Dos de estos datos son los booleanos y los de tipo fecha.

* Las propiedades **booleanas** son las que pueden tener un valor de Verdadero o Falso, por ejemplo, la propiedad Exists del comando Get-ChildItem.

****

Cuando queremos filtrar por estas propiedades y queremos poner que un valor es verdadero o falso, no podemos poner directamente True o False, ya que el sistema las interpretará como cadenas de texto en lugar de hacerlo como valores booleanos. En estos casos, es necesario utilizar dos variables del sistema que representaremos de la forma **$True** y **$False**.

* Otro tipo de propiedades muy común son las de fecha y hora, que podemos encontrar por ejemplo en la fecha de creación de un fichero.

**Texto

Descripción generada automáticamente**

* Aquí encontramos el mismo problema que en el caso anterior ya que si ponemos la fecha directamente la interpretará como una cadena. En este caso, hay que utilizar el comando **Get-Date** con el parámetro **-date** que convierte una fecha en modo texto a un objeto de tipo datetime que almacena dicha fecha.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Pero ahora hay otro problema, ¿cómo hacemos para incluir el valor devuelto por este comando en el parámetro de otro comando? En este caso tenemos que recurrir a los paréntesis de la siguiente forma:



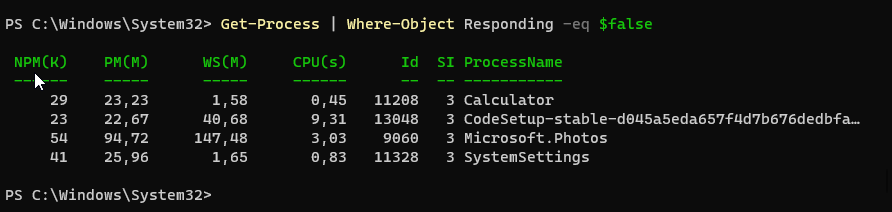
Los **paréntesis** hacen que en primer lugar se ejecute el comando que hay en su interior y, el valor devuelto por dicho comando reemplazará todo lo que hay entre paréntesis.

Hay diversas formas de indicar la fecha que se le pasa al comando Get-Date, tanto con fecha y hora como solo fecha. Algunos ejemplos son:

* + “2 de noviembre de 2021 10:05:00”
  + “02/11/2021”
  + “02/11/21 10:10:30”
  + “2021-02-11”

Teniendo en cuenta lo anterior, realiza los siguientes ejercicios:

**10.-** Muestra todos los procesos que tienen el estado Respond puesto a False, es decir, todos los procesos del sistema que se hayan colgado.



**11.-** Muestra todos los ficheros de C:\Windows que hayan sido creados con fecha posterior al 15 de octubre.

