

PR0302: Introducción a Powershell (II)

Nombre/s: Diego Fernández Pita

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 renombra 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.

```
PS C:\Users\Diego> Get-Date -Format yyyy
2022

PS C:\WINDOWS\system32> Get-Date | Select-Object year
Year
----
2022
```

2.- Uno de los requisitos de Windows 11 es que el 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**.

```
PS C:\WINDOWS\system32> Get-Tpm | Select-Object TpmPresent, TpmReady, TpmEnabled, TpmActivated | Format-Table

TpmPresent TpmReady TpmEnabled TpmActivated
-----
True       True       True       True
```

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.

```
PS C:\WINDOWS\system32> Get-ChildItem | measure

Count      : 4845
Average    :
Sum        :
Maximum    :
Minimum    :
Property   :
```

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**.

```
PS C:\WINDOWS\system32> Get-ChildItem *.dll | measure_
```

Count	: 3618
Average	:
Sum	:
Maximum	:
Minimum	:
Property	:

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

```
PS C:\WINDOWS\system32> Get-ChildItem | Where-Object *.exe -gt 50000 | measure
```

Count	: 0
Average	:
Sum	:
Maximum	:
Minimum	:
Property	:

Para demostrar que la búsqueda es correcta, incluyo la búsqueda al revés...

```
PS C:\WINDOWS\system32> Get-ChildItem | Where-Object *.exe -lt 50000 | measure
```

Count	: 4845
Average	:
Sum	:
Maximum	:
Minimum	:
Property	:

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*).

```
PS C:\WINDOWS\system32> Get-ChildItem *.dll | Select-Object -Property CreationTime, LastAccessTime, Name | Sort-Object CreationTime
```

CreationTime	LastAccessTime	Name
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100ita.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	vcomp100.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100fra.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100jpn.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100rus.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100kor.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100cht.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100chs.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100u.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100esn.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100enu.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100deu.dll
11/06/2011 2:15:38	13/10/2022 16:04:42	msvc100.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	atl100.dll
11/06/2011 2:15:38	13/10/2022 16:04:42	msvcr100.dll
11/06/2011 2:15:38	13/10/2022 16:04:44	mfc100u.dll
05/11/2012 23:26:22	13/10/2022 16:04:42	vccorlib110.dll
05/11/2012 23:26:22	13/10/2022 16:04:43	vcamp110.dll
05/11/2012 23:26:22	13/10/2022 16:04:42	msvcr110.dll
05/11/2012 23:26:22	13/10/2022 16:04:42	msvc110.dll
05/11/2012 23:26:22	13/10/2022 16:04:43	vcomp110.dll
09/06/2013 16:59:36	13/10/2022 16:04:43	mfc110rus.dll
09/06/2013 16:59:36	13/10/2022 16:04:43	mfc110u.dll
09/06/2013 16:59:36	13/10/2022 16:04:42	mfc110.dll

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

```
PS C:\WINDOWS\system32> Get-ChildItem | Select-Object -Property Length, FullName | Sort-Object -Descending

Length FullName
-----
123392 C:\WINDOWS\system32\ReportingCSP.dll
250696 C:\WINDOWS\system32\RESAMPLEDMO.DLL
129024 C:\WINDOWS\system32\repair-bde.exe
22528 C:\WINDOWS\system32\replace.exe
113152 C:\WINDOWS\system32\ResBParser.dll
```

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.

```
PS C:\WINDOWS\system32> Get-ChildItem | Select-Object Length, FullName | Where-Object Length -gt 10000000 | Sort-Object length -Descending

Length FullName
-----
147398024 C:\WINDOWS\system32\MRT.exe
40346984 C:\WINDOWS\system32\nvcompiler.dll
32608744 C:\WINDOWS\system32\WindowsCodecsRaw.dll
31244248 C:\WINDOWS\system32\nvopengl.dll
26268672 C:\WINDOWS\system32\edgehtml.dll
25076000 C:\WINDOWS\system32\mfplugin64_hw.dll
24272384 C:\WINDOWS\system32\Hydrogen.dll
23449600 C:\WINDOWS\system32\mshtml.dll
21448544 C:\WINDOWS\system32\libmfhw64.dll
18767872 C:\WINDOWS\system32\HologramWorld.dll
17750352 C:\WINDOWS\system32\nvcuda.dll
17551872 C:\WINDOWS\system32\Windows.UI.Xaml.dll
14208384 C:\WINDOWS\system32\vmms.exe
13728120 C:\WINDOWS\system32\nvptxjitCompiler.dll
11445248 C:\WINDOWS\system32\wmp.dll
10846592 C:\WINDOWS\system32\ntoskrnl.exe
10349360 C:\WINDOWS\system32\Windows.Media.PlayReady.dll
```

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.

```
PS C:\WINDOWS\system32> Get-ChildItem *.exe | Select-Object Length, FullName | Where-Object Length -gt 10000000 | Sort-Object length -Descending

Length FullName
-----
147398024 C:\WINDOWS\system32\MRT.exe
14208384 C:\WINDOWS\system32\vmms.exe
10846592 C:\WINDOWS\system32\ntoskrnl.exe
```

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`.

DirectoryName	Property	string	DirectoryName {get
Exists	Property	bool	Exists {get;}
Extension	Property	string	Extension {get;}

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.

```
PS C:\Users\victor> Get-ChildItem | Get-Member CreationTime

TypeName: System.IO.DirectoryInfo

Name      MemberType Definition
-----
CreationTime Property      datetime CreationTime {get;set;}
```

- 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.

```
PS C:\Users\victor> get-date -date "2 de noviembre de 2021"

martes, 2 de noviembre de 2021 0:00:00
```

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:

```
PS C:\Users\victor> Get-ChildItem | Where-Object CreationTime -gt (Get-Date -date "1 de octubre de 2021")
```

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.

```
PS C:\WINDOWS\system32> Get-Process | Where-Object Responding -eq $false
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
1053	48	42456	1652	3,55	20540	3	HxOutlook
1161	50	28072	1448	2,70	11548	3	SystemSettings

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

```
PS C:\WINDOWS> Get-ChildItem | Where-Object CreationTime -gt (Get-Date -Date "15 de octubre de 2022")
PS C:\WINDOWS> Get-ChildItem | Where-Object CreationTime -gt (Get-Date -Date "15 de octubre de 2022")

Directorio: C:\WINDOWS

Mode                LastWriteTime         Length Name
----                -
d-----         18/10/2022    13:01             Nueva carpeta
```

Como no aparecía ninguna carpeta posterior al 15 de octubre de 2022, he optado por crear una para mostrarla.