

Ge



**Ciclo:** Administración de Sistemas Informáticos en Red  
**Módulo:** IMPLANTACIÓN DE SISTEMAS OPERATIVOS  
**Grupo:** 1º ASIR

## PR0302: Introducción a Powershell (II)

**Nombre/s:** Luis Eduardo Escobar Alegría

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} – PR0201**. 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.- Visualiza las últimas cinco entradas del historial, mostrando para cada una el comando, la hora en que finalizó su ejecución y el estado de ejecución.

```
PS C:\Users\luise> Get-History -count 5 | Select-Object EndExecutionTime, CommandLine, ExecutionStatus
```

EndExecutionTime	CommandLine	ExecutionStatus
11/10/2022 13:53:44	Get-Service   Group-Object Status	Completed
11/10/2022 13:54:46	Get-Service   Group-Object Status, Startype	Completed
11/10/2022 13:54:57	Get-Service	Completed
11/10/2022 14:00:57	Get-History   Get-Member	Completed
11/10/2022 14:04:55	Get-History   Select-Object -id 6,8,9	Completed

2.- Ejecuta el comando **Get-Command** (que muestra todos los comandos disponibles en Powershell) e interrúmpelo antes de que finalice su ejecución pulsando las teclas Ctrl-C. A continuación, ejecútalo dejando que finalice correctamente.

```
comando. la hora en que finalizó su ejecución y el estado de ejecución.
```

```
PowerShell 7 (x64)
```

Cmdlet	Write-Output	7.0.0.0	Microsoft.PowerShell.Utility
Cmdlet	Write-Progress	7.0.0.0	Microsoft.PowerShell.Utility
Cmdlet	Write-Verbose	7.0.0.0	Microsoft.PowerShell.Utility
Cmdlet	Write-Warning	7.0.0.0	Microsoft.PowerShell.Utility

```
PS C:\Users\luise> Get-Command
```

CommandType	Name	Version	Source
Alias	Add-AppPackage	2.0.1.0	Appx
Alias	Add-AppPackageVolume	2.0.1.0	Appx
Alias	Add-AppProvisionedPackage	3.0	Dism
Alias	Add-ProvisionedAppPackage	3.0	Dism
Alias	Add-ProvisionedAppSharedPackageContainer	3.0	Dism
Alias	Add-ProvisionedAppxPackage	3.0	Dism
Alias	Add-ProvisioningPackage	3.0	Provisioning
Alias	Add-TrustedProvisioningCertificate	3.0	Provisioning
Alias	Apply-WindowsUnattend	3.0	Dism
Alias	Disable-PhysicalDiskIndication	2.0.0.0	Storage
Alias	Disable-PhysicalDiskIndication	1.0.0.0	VMDirectStorage
Alias	Disable-StorageDiagnosticLog	2.0.0.0	Storage
Alias	Disable-StorageDiagnosticLog	1.0.0.0	VMDirectStorage
Alias	Dismount-AppPackageVolume	2.0.1.0	Appx
Alias	Enable-PhysicalDiskIndication	2.0.0.0	Storage
Alias	Enable-PhysicalDiskIndication	1.0.0.0	VMDirectStorage
Alias	Enable-StorageDiagnosticLog	2.0.0.0	Storage
Alias	Enable-StorageDiagnosticLog	1.0.0.0	VMDirectStorage
Alias	Export-VMCheckpoint	2.0.0.0	Hyper-V
Alias	Flush-Volume	2.0.0.0	Storage
Alias	Flush-Volume	1.0.0.0	VMDirectStorage

3.- Vuelve a ejecutar el comando del punto 1 y comprueba las diferentes salidas de finalización de estado de ejecución.

```
PS C:\Users\luise> Get-History | Select-Object EndExecutionTime
```

```
EndExecutionTime
```

```
-----
```

```
11/10/2022 13:52:58
```

```
11/10/2022 13:53:44
```

```
11/10/2022 13:54:46
```

```
11/10/2022 13:54:57
```

```
11/10/2022 14:00:57
```

```
11/10/2022 14:04:55
```

```
11/10/2022 14:06:09
```

```
11/10/2022 14:07:04
```

```
11/10/2022 14:07:25
```

```
11/10/2022 14:07:36
```

```
11/10/2022 14:07:50
```

```
11/10/2022 14:08:22
```

```
11/10/2022 14:08:41
```

```
11/10/2022 14:08:50
```

```
11/10/2022 14:09:18
```

```
11/10/2022 14:09:31
```

```
PS C:\Users\luise>
```

4.- Muestra todos los procesos con el nombre *msedge* mostrando para cada uno el identificador, el consumo de CPU y los hilos (*threads*)

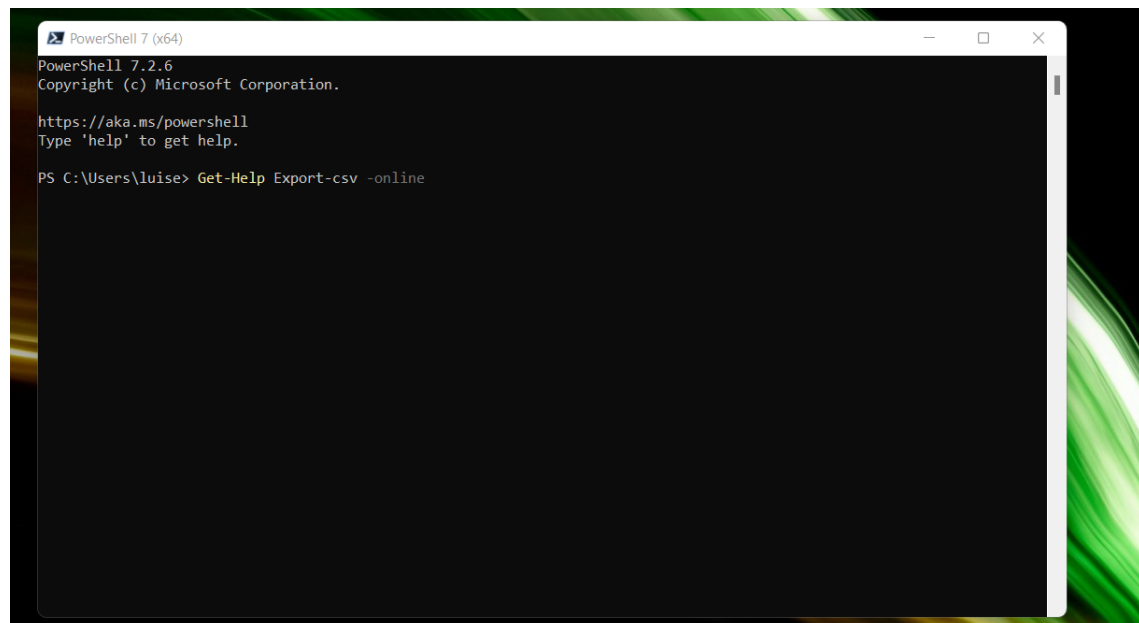
```
PS C:\Users\luise> Get-Process msedge | Select-Object Id, CPU, threads

Id      CPU Threads
--      -
1304 0,90625 {6660, 10744, 6324, 840...}
3496 0,0625 {3536, 5088, 12124, 11980...}
8536 4,84375 {10180, 6468, 11844, 3540...}
9292 0,09375 {15872, 8824, 15296, 9656...}
9896 0,34375 {6196, 14064, 9908, 1744...}

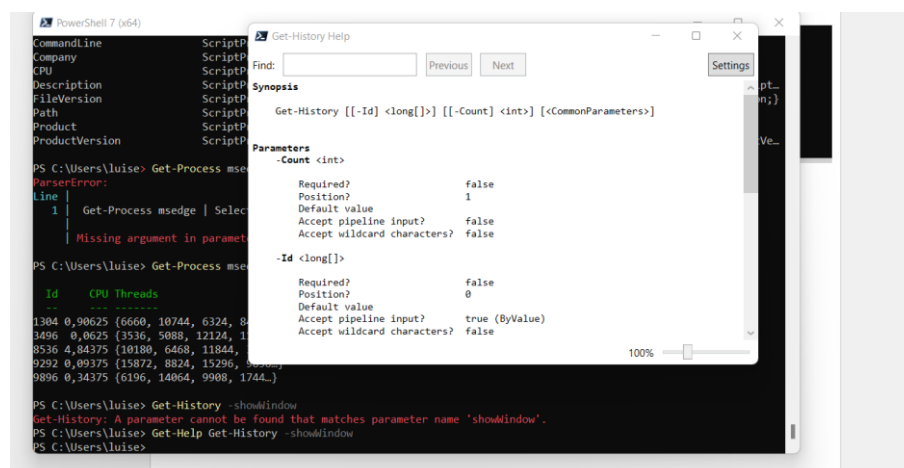
PS C:\Users\luise>
```

5.- Averigua para qué sirve el parámetro **-Delimiter** del comando **Export-CSV**

El parámetro *Delimiter* especifica un punto y coma para separar los valores de cadena. El parámetro *NoTypeInformation* quita el encabezado de información *#TYPE* de la salida CSV y no es necesario en PowerShell 6.



6.- Muestra en una ventana la ayuda del comando **Get-History**



7.- Muestra un listado con todos los comandos que tengan el verbo *Update*.

```
PowerShell 7 (x64)
PS C:\Users\luise> Get-Command -verb Update

CommandType Name Version Source
-----
Function Update-AutologgerConfig 1.0.0.0 EventTracingManagement
Function Update-Disk 2.0.0.0 Storage
Function Update-DscConfiguration 1.1 PSDesiredStateConfiguration
Function Update-EtwTraceSession 1.0.0.0 EventTracingManagement
Function Update-HostStorageCache 2.0.0.0 Storage
Function Update-IscsiTarget 1.0.0.0 iSCSI
Function Update-IscsiTargetPortal 1.0.0.0 iSCSI
Function Update-Module 2.2.5 PowerShellGet
Function Update-Module 1.0.0.1 PowerShellGet
Function Update-ModuleManifest 2.2.5 PowerShellGet
Function Update-ModuleManifest 1.0.0.1 PowerShellGet
Function Update-MpSignature 1.0 ConfigDefender
Function Update-MpSignature 1.0 Defender
Function Update-NetFirewallDynamicKeywordAddress 2.0.0.0 NetSecurity
Function Update-NetIPsecRule 2.0.0.0 NetSecurity
Function Update-Script 2.2.5 PowerShellGet
Function Update-Script 1.0.0.1 PowerShellGet
Function Update-ScriptFileInfo 2.2.5 PowerShellGet
Function Update-ScriptFileInfo 1.0.0.1 PowerShellGet
Function Update-SmbMultichannelConnection 2.0.0.0 SmbShare
Function Update-StorageBusCache 1.0.0.0 StorageBusCache
Function Update-StorageFirmware 2.0.0.0 Storage
Function Update-StoragePool 2.0.0.0 Storage
Function Update-StorageProviderCache 2.0.0.0 Storage
Cmdlet Update-FormatData 7.0.0.0 Microsoft.PowerShell.Utility
Cmdlet Update-Help 7.2.6.500 Microsoft.PowerShell.Core
Cmdlet Update-List 7.0.0.0 Microsoft.PowerShell.Utility
Cmdlet Update-TypeData 7.0.0.0 Microsoft.PowerShell.Utility
Cmdlet Update-VMVersion 2.0.0.0 Hyper-V
Cmdlet Update-WIMBootEntry 3.0 Dism

PS C:\Users\luise>
```

8.- Ejecuta la herramienta *Recortes* y localízala usando el comando **Get-Process** teniendo en cuenta que el proceso se llama *SnippingTool.exe*

```
CommandType Name Version Source
-----
Application SnippingTool.exe 0.0.0.0 C:\Users\luise\AppData\Local\Microsoft\Win
dowsApps\SnippingTool.exe

PS C:\Users\luise>
```

9.- Averigua qué propiedades tienen los procesos devueltos con el comando **Get-Process**.

```
PowerShell 7 (x64)
C:\WindowsApps\SnippingTool.exe

PS C:\Users\luise> Get-Process | Get-Member

TypeName: System.Diagnostics.Process

Name                MemberType          Definition
-----
Handles             AliasProperty      Handles = Handlecount
Name                AliasProperty      Name = ProcessName
NPM                 AliasProperty      NPM = NonpagedSystemMemorySize64
PM                  AliasProperty      PM = PagedMemorySize64
SI                  AliasProperty      SI = SessionId
VM                  AliasProperty      VM = VirtualMemorySize64
WS                  AliasProperty      WS = WorkingSet64
Parent              CodeProperty        System.Object Parent{get=GetParentProcess;}
Disposed            Event               System.EventHandler Disposed(System.Object, System.EventArgs)
ErrorDataReceived   Event               System.Diagnostics.DataReceivedEventHandler ErrorDataReceived(System.Object,...
Exited              Event               System.EventHandler Exited(System.Object, System.EventArgs)
OutputDataReceived  Event               System.Diagnostics.DataReceivedEventHandler OutputDataReceived(System.Object...
BeginErrorReadLine  Method              void BeginErrorReadLine()
BeginOutputReadLine Method              void BeginOutputReadLine()
CancelErrorRead     Method              void CancelErrorRead()
CancelOutputRead    Method              void CancelOutputRead()
Close               Method              void Close()
CloseMainWindow     Method              bool CloseMainWindow()
Dispose             Method              void Dispose(), void IDisposable.Dispose()
Equals              Method              bool Equals(System.Object obj)
GetHashCode          Method              int GetHashCode()
```

10.- Busca en la ayuda para qué sirve el parámetro **-MemberType** del comando **Get-Member**.

```
Get-Help: No parameter matches criteria 'MemberType'.
PS C:\Users\luise> Get-Help Get-Member -Parameter memberType

-MemberType <PSMemberTypes>

Required?           false
Position?           Named
Accept pipeline input? false
Parameter set name   (All)
Aliases             Type
Dynamic?            false
Accept wildcard characters? false

PS C:\Users\luise>
```

11.- Desde la línea de comandos, finaliza la ejecución de la herramienta *recortes*.

```
PS C:\Users\luise> Get-Process | SnippingTool | Stop-Process
PS C:\Users\luise>
```

12.- Muestra todos los procesos que tienen el nombre *svchost*.

```
PowerShell 7 (x64)
PS C:\Users\luise> Get-Process svchost
```

NPM(K)	PM(M)	WS(M)	CPU(s)	Id	SI	ProcessName
26	17,48	39,70	0,00	100	0	svchost
20	2,72	10,44	0,00	732	0	svchost
18	4,36	25,49	3,05	960	10	svchost
21	12,46	18,70	0,00	1140	0	svchost
13	3,47	9,07	0,00	1204	0	svchost
8	1,10	4,81	0,00	1356	0	svchost
11	1,89	8,55	0,00	1416	0	svchost
13	2,52	9,88	0,00	1436	0	svchost
10	2,43	11,89	0,00	1448	0	svchost
7	2,78	6,01	0,00	1456	0	svchost
20	7,44	16,64	0,00	1560	0	svchost
12	2,73	8,56	0,00	1632	0	svchost
11	6,33	9,73	0,00	1652	0	svchost
32	7,09	10,33	0,00	1696	0	svchost
10	2,82	8,39	0,00	1732	0	svchost
20	6,51	16,54	0,00	1800	0	svchost
12	3,41	11,53	0,00	1904	0	svchost
8	1,36	6,14	0,00	1924	0	svchost
14	2,35	9,27	0,00	1968	0	svchost
15	4,32	21,12	0,00	2032	0	svchost
13	2,70	9,51	0,00	2260	0	svchost
16	3,58	8,96	0,00	2296	0	svchost
17	11,31	20,30	0,00	2324	0	svchost
11	3,46	7,72	0,00	2508	0	svchost
15	18,50	18,48	0,00	2552	0	svchost
13	3,73	11,27	0,00	2560	0	svchost

13.- Muestra por pantalla el número de instancias del proceso *svchost*.

```
PowerShell 7 (x64)
PowerShell 7.2.6
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\luise> Get-Process -name svchost | Measure-Object
```

Count	: 81
Average	:
Sum	:
Maximum	:
Minimum	:
StandardDeviation	:
Property	:

```
PS C:\Users\luise>
```

14.- Muestra por pantalla todos los procesos con el nombre *svchost* mostrando para cada uno: nombre, identificador, hora de inicio, tiempo total de procesador y clase de prioridad. Se deben mostrar de forma tabular.

```
PS C:\Users\luise> Get-Process svchost | Select-Object Name, Id, StartTime, TotalProcessorTime, PriorityClass | Format-Table
```

Name	Id	StartTime	TotalProcessorTime	PriorityClass
svchost	100			
svchost	732			
svchost	960	13/10/2022 9:36:20	00:00:03.3750000	Normal
svchost	1140			
svchost	1204			
svchost	1356			
svchost	1416			
svchost	1436			
svchost	1448			
svchost	1456			
svchost	1560			
svchost	1632			
svchost	1652			
svchost	1696			
svchost	1732			
svchost	1800			
svchost	1904			
svchost	1924			
svchost	1968			
svchost	2032			
svchost	2260			
svchost	2296			
svchost	2324			

15.- Repite la búsqueda anterior, pero ordenando por el campo *tiempo total de procesador* en sentido descendente.

```
PS C:\Users\luise> Get-Process svchost | Select-Object Name, Id, StartTime, TotalProcessorTime, PriorityClass | sort-object totalprocesortime -descending
```

Name	Id	StartTime	TotalProcessorTime	PriorityClass
svchost	8164	13/10/2022 9:36:19	00:00:09.6406250	Normal
svchost	960	13/10/2022 9:36:20	00:00:04.1250000	Normal
svchost	11900	13/10/2022 9:36:19	00:00:03.4062500	Normal
svchost	14576			

16.- Muestra los usuarios que hay en el sistema agrupándolos por la propiedad *Enabled*.

```
PowerShell 7.2.6
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

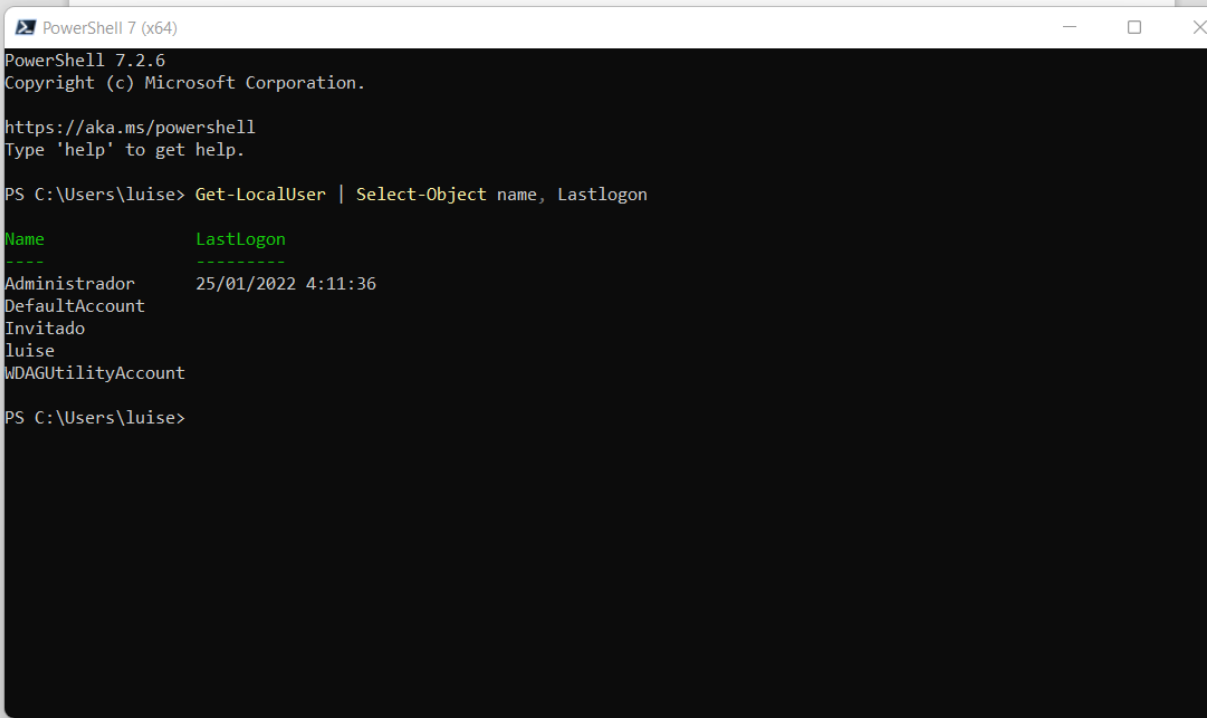
PS C:\Users\luise> Get-LocalUser | Group-Object enabled
```

Count	Name	Group
4	False	{Administrador, DefaultAccount, Invitado, WDAGUtilityAccount}
1	True	{luise}

```
PS C:\Users\luise>
```

**17.-** Muestra los usuarios que hay en el sistema con la cuenta habilitada (propiedad *Enabled* puesta a *True*). Utiliza el filtrado con el comando **Where-Object**

**18.-** Muestra un listado de todos los usuarios del sistema con el nombre y la fecha de la última vez que iniciaron sesión (tienes que buscar la propiedad que indique último inicio de sesión o *last logon*)



```
PowerShell 7 (x64)
PowerShell 7.2.6
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\luise> Get-LocalUser | Select-Object name, LastLogon

Name                LastLogon
----                -
Administrador       25/01/2022 4:11:36
DefaultAccount
Invitado
luise
WDAGUtilityAccount

PS C:\Users\luise>
```