

## Sesión 1 - Fundamentos de PowerShell (I)

Alumno 1 Rubén Martínez Ginzo

Alumno 2 Juan Francisco Mier Montoto

**(1) EJERCICIO.** ¿Qué comando debes ejecutar para listar todos los servicios que incluyan la palabra “redes” en su *DisplayName*? Indícalo a continuación:

```
Get-Service -DisplayName *redes*
```

**(2) EJERCICIO.** Ejecuta un comando que liste todos los comandos de tipo CmdLet. Indica el comando a continuación.

```
Get-Service -CommandType CmdLet
```

**(3) EJERCICIO.** Ejecuta un comando que liste todos los comandos de tipo CmdLet que contengan la palabra *Process*. Indica el comando a continuación.

```
Get-Service -CommandType CmdLet -Name *process*
```

**(4) EJERCICIO.** Utiliza el parámetro *-Verb* para listar los comandos que utilizan el verbo *Get*. Indica el comando a continuación.

```
GetCommnad -Verb Get
```

**(5) EJERCICIO.** Utiliza el comando *Get-Help* para analizar el funcionamiento del comando *Write-Host*. Después, ejecuta este comando de modo que escriba en la consola la cadena de caracteres “Prácticas de PowerShell”, utilizando fuente de color rojo y fondo de color amarillo. Indica el comando ejecuta a continuación.

```
Write-Host -ForegroundColor red -BackgroundColor yellow  
"Prácticas de PowerShell"
```

**(6) EJERCICIO.** Utilizando el parámetro *-Separator*, escribe un comando *Write-Host* que imprima en la consola las cadenas de caracteres “Pedro”, “María” y “Juan” separadas mediante el carácter ‘/’. Las cadenas deben pasarse al comando separadas por comas.

```
Write-Host -Separator "/" Pedro, María, Juan
```

**(7) EJERCICIO.** Utilizando previamente la ayuda de *Get-Command*, ejecuta este comando de modo que muestre una lista de todos los Alias disponibles en la consola de PowerShell. Indica a continuación el comando que has ejecutado.

```
Get-Command -CommandType Alias
```

**(8) EJERCICIO.** Se desea utilizar el comando *Get-Alias* para conocer el alias correspondiente al comando *Get-Command*, si existiera. Utiliza la ayuda para ver cómo debes ejecutar el comando *Get-Alias*. Escribe a continuación el comando requerido.

```
Get-Alias -Definition Get-Command
```

**(9) EJERCICIO.** Ejecuta un comando *gcm* para obtener un listado de todos los comandos que actúen sobre el nombre *Host*. Indica a continuación el comando que has ejecutado.

```
Gcm -Noun Host
```

**(10) EJERCICIO.** Ejecuta una línea de comandos que solicite al usuario introducir una cadena de caracteres por consola y que la imprima en color rojo. La línea de comandos debe generar el mensaje Introduce una cadena: , con objeto de guiar al usuario en la acción a realizar.

- Para hacer este ejercicio necesitas utilizar un comando aún no visto. Se trata de *Read-Host*. Consulta la ayuda para aprender a utilizarlo.
- Este ejercicio requiere ejecutar dos comandos. Para encadenar comandos en una misma línea se utiliza el carácter ‘;’.

```
$a = Read-Host "Introduce una cadena"  
Write-Host ForegroundColor Red $a
```

**(11) EJERCICIO.** Ejecuta una línea de comandos que solicite al usuario introducir una cadena de caracteres por consola y que, a continuación, reemplace todos los caracteres ‘a’ por el carácter ‘%’. Para ello, será necesario definir una variable de tipo cadena, utilizar un método del objeto tipo *System.String* y encadenar dos comandos mediante el operador ‘;’. Escribe la línea de comandos ejecutada a continuación.

```
$a = Read-Host "Introduce una cadena"  
$a.Replace('a', '%')
```

**(12) EJERCICIO.** Define una variable que se llame \$Pi e inicialízala con el valor 3.1416. Después, utilizando el comando *Get-Member*, determina el tipo de objeto asignado a esta variable. Indica el tipo a continuación.

```
System.Double
```

**(13) EJERCICIO.** Tras analizar la información proporcionada por *Get-Member*, ejecuta un comando de PowerShell que muestre el PID del proceso *Notepad*. Comprueba que el valor proporcionado por PowerShell coincide con el indicado por el *Administrador de tareas*. Escribe el comando ejecutado a continuación.

```
(Get-Process notepad).Id
```

**(14) EJERCICIO.** Ejecuta un comando de PowerShell que muestre la fecha y hora en la que ha comenzado a ejecutarse el proceso *Notepad*. Comprueba que el valor proporcionado por PowerShell es coherente con el momento aproximado de arranque del proceso. Escribe el comando a continuación.

```
(Get-Process notepad).StartTime
```

**(15) EJERCICIO.** Ejecuta el método *Kill()* para eliminar el proceso de ejecución. Comprueba que *Notepad* se cierra y que el proceso es eliminado de la lista de procesos mostrados por el *Administrador de tareas*. Escribe el comando ejecutado a continuación.

```
(Get-Process notepad).kill()
```

**(16) EJERCICIO.** Hasta ahora, se ha utilizado *Get-Member* pasándole los objetos de entrada mediante el parámetro *-InputObject*. *Get-Member* también puede recibir los objetos de entrada a través de una canalización. Así, para obtener todas las propiedades y métodos de un objeto proceso, ejecuta el comando *Get-Process* pasándole el nombre de un proceso cualquiera y canaliza su salida hacia *Get-Member*. Escribe el comando a continuación.

**Get-Process winlogon | Get-Member**

**(17) EJERCICIO.** Ejecuta una línea de comandos que muestre los procesos en ejecución en formato de lista, indicando de cada uno de ellos las siguientes propiedades: Nombre del proceso, fecha y hora en la que ha comenzado a ejecutarse el proceso y ruta en la que se encuentra el fichero ejecutable del proceso. Escribe la línea de comandos ejecutada a continuación.

**Get-Process | Format-List -Property name,starttime,path**

**(18) EJERCICIO.** Repite el ejercicio anterior, pero mostrando los procesos (con las tres propiedades indicadas) en formato de tabla. Escribe la línea de comandos ejecutada a continuación.

**Get-Process | Format-Table -Property name,starttime,path**

**(19) EJERCICIO.** Ejecutar una línea de comandos que muestre todos los procesos del sistema cuyo nombre empiece por 's' y ordenados de forma descendente por su 'Id'. Escribe el comando a continuación.

**Get-Process s\* | Sort-Object Id -Descending**

**(20) EJERCICIO.** El comando *Get-Alias* proporciona los Alias definidos en la consola de PowerShell. Por defecto, este comando ordena su información de salida por el nombre de los Alias. Ejecuta una línea de comandos que muestre los alias definidos en la consola de PowerShell ordenados por su definición, en vez de por su nombre. Escribe la línea de comando ejecutada a continuación.

**Get-Alias | Sort-Object Definition**

**(21) EJERCICIO.** Obtén ayuda del comando *Get-PSDrive*. Después, ejecuta este comando de modo que muestre solo las unidades gestionadas por el proveedor de tipo *Filesystem*. Escribe el comando ejecutado a continuación.

**Get-PSDrive -PSProvider FileSystem**

**(22) EJERCICIO.** Obtén ayuda del comando *Get-ChildItem*. Determina el parámetro necesario para indicar la ubicación de trabajo. Después, ejecuta este comando de modo que obtenga todos los archivos que se encuentren en la ubicación *\Windows\System32* que sean de tipo ejecutable, es decir, que tengan la extensión *.exe*. Además, se debe conseguir que la salida se muestre página a página. Indica el comando requerido a continuación.

**Get-ChildItem -Path \Windows\System32\\*.exe | Out-Host -Paging**

**(23) EJERCICIO.** Busca los alias definidos para el comando *Get-ChildItem*. Para ello, obtén ayuda del comando *Get-Alias* y fíjate en el parámetro *-Definition*. Indica el comando requerido a continuación.

**Get-Alias -Definition get-ChildItem**

**(24) EJERCICIO.** Solicita ayuda sobre el comando *New-Item*. Después, utilizando este comando crea un directorio llamado *Temp* en el directorio raíz del volumen C. A continuación, ejecuta otro comando que posicione la consola en el directorio que acabas de crear. Finalmente, ejecuta otro comando que cree en *Temp* un fichero de texto llamado *prueba.txt* que contenga el texto “Esto es una prueba”. NOTA: No utilizar el parámetro *-Path* para que las operaciones realizadas por *New-Item* tengan lugar sobre el directorio (posición) actual. Indica los comandos requeridos a continuación.

```
New-Item -Type Directory Temp
cd .\Temp\
New-Item -Type File prueba.txt
Set-Content .\prueba.txt -Value “Esto es una prueba”
```

**(25) EJERCICIO.** Utilizando el comando *Remove-Item*, elimina el fichero *prueba.txt* y el directorio *Temp*. Indica los comandos requeridos a continuación.

```
Remove-Item .\prueba.txt
cd ..
Remove-Item .\Temp\
```