



# Certified Tech Developer

The Ultimate Degree

preview

## Infraestructura I

# Glosario de comandos

A continuación, veremos algunos de los cmdlets más útiles y más comunes que estaremos usando en Windows PowerShell.

25 comandos básicos de Windows PowerShell

Cmdlet	Descripción.
Set-Location	Establece la ubicación de trabajo actual en una ubicación específica.
Get-Content	Obtiene el contenido del elemento en la ubicación especificada.
Add-Content	Agrega contenido a los elementos especificados, por ejemplo, agregar palabras a un archivo.
Set-Content	Escribe o reemplaza el contenido de un elemento con contenido nuevo.
Copy-Item	Copia un elemento de una ubicación a otra.
Remove-Item	Elimina los elementos especificados.
Move-Item	Mueve un elemento de una ubicación a otra.
Set-Item	Cambia el valor de un elemento al valor especificado en el comando.
New-Item	Crea un nuevo elemento.
Start-Job	Inicia un trabajo en segundo plano de Windows PowerShell.

Compare-Object	Compara dos conjuntos de objetos.
Group-Object	Agrupar objetos que contengan el mismo valor en propiedades especificadas.
Invoke-WebRequest	Obtiene contenido de una página web en Internet.
Measure-Object	Calcula los valores de propiedad de determinados tipos de objetos. Puede realizar tres tipos de mediciones según los parámetros del comando.
Resolve-Path	Resuelve los caracteres comodín en una ruta y muestra el contenido.
Resume-Job	Reinicia un trabajo suspendido.
Set-Variable	Establece el valor de una variable. Crea la variable si no existe una con el nombre solicitado.
Show-Command	Muestra comandos de Windows PowerShell en una ventana de comandos gráfica.
Sort-Object	Ordena los objetos por valores de propiedad.
Start-Service	Inicia uno o más servicios detenidos.
Start-Process	Inicia uno o más procesos en la computadora local.
Suspend-Job	Detiene temporalmente un trabajo.
Wait-Job	Espera a que los trabajos en segundo plano de Windows PowerShell se completen antes de mostrar el símbolo de sistema.
Where-Object	Selecciona los objetos del conjunto de objetos que se le pasan.
Write-Output	Envía el objeto especificado al siguiente comando en la canalización. Si es el último comando en la canalización, el objeto se mostrará en consola.

## Ejemplos de uso

- Set-Location: Establece la ubicación actual.

```
PS C:\> Set-Location -Path "HKLM:\"  
  
PS HKLM:\>
```

- Get-Content: Obtiene el contenido de un archivo de texto.

```
Get-Content -Path .\LineNumbers.txt  
  
Esta es la línea 1  
  
Esta es la línea 2
```

- Add-Content: Añade contenido a un archivo de texto.

```
Add-Content -Path .\LineNumbers.txt -Value 'Fin de archivo'  
  
Esta es la línea 1  
  
Esta es la línea 2  
Fin de archivo
```



- Set-Content: Modifica una línea de un archivo de texto.

```
(Get-Content -Path .\LineNumbers.txt) |  
    ForEach-Object {$_ -Replace 'Fin de archivo', 'Esta es la línea 3'}  
|  
    Set-Content -Path .\LineNumbers.txt  
  
Get-Content -Path .\LineNumbers.txt  
  
Esta es la línea 1  
  
Esta es la línea 2  
  
Esta es la línea 3
```

- Copy-Item: Copia un archivo a un directorio específico.

```
Copy-Item "C:\Wabash\Logfiles\mar1604.log.txt" -Destination  
"C:\Presentation"
```

- Remove-Item: Este ejemplo elimina de la carpeta actual todos los archivos que tienen una extensión de nombre de archivo .doc con la excepción de los que tengan un \*1\* en su nombre.

```
Remove-Item * -Include *.doc -Exclude *1*
```

- Move-Item: Mueve un archivo a otro directorio y lo renombra.

```
Move-Item -Path C:\test.txt -Destination E:\Temp\tst.txt
```

- Set-Item: Este ejemplo crea un alias llamado np para el Notepad de Windows.

```
Set-Item -Path alias:np -Value "c:\windows\notepad.exe"
```

- New-Item: Crea un archivo en el directorio actual.

```
New-Item -Path . -Name "archivo.txt" -ItemType "file" -Value "Texto de prueba"
```

- Start-Job: Ejecuta un script como un proceso en segundo plano.

```
Start-Job -FilePath C:\Scripts\Sample.ps1
```

- Compare-Object: Compara el contenido de dos archivos:

Este ejemplo compara el contenido de dos archivos, el ejemplo usa los siguientes dos archivos de texto, cada uno con un valor en una línea diferente.

- verduras.txt contiene los valores: papa, zapallo, tomate.
- frutas.txt contiene los valores: banana, tomate, naranja.

El resultado muestra solo las líneas que son diferentes entre los archivos. En verduras.txt usa el objeto de referencia (<=) y en frutas.txt usa el objeto de diferencia (=>). Las líneas que se encuentran en los dos archivos no son mostradas.

```
Compare-Object -ReferenceObject (Get-Content -Path C:\Test\verduras.txt)
-DifferenceObject (Get-Content -Path C:\Test\frutas.txt)
```

```
InputObject SideIndicator
```

```
-----
```

```
banana      =>
```

```
naranja     =>
```

```
papa        <=
```

```
zapallo     <=
```

- Group-Object: Agrupa archivos por su extensión.

```
$files = Get-ChildItem -Path $PSHOME -Recurse
$files | Group-Object -Property extension -NoElement | Sort-Object -
Property Count -Descending
```

```
Count Name
```

```
-----
```

```
365 .xml
```

```
231 .cdxml
```

```
197
```

```
169 .ps1xml
```



```
142 .txt  
114 .psd1  
63 .psm1  
49 .xsd  
36 .dll  
15 .mfl  
15 .mof  
...
```

- Invoke-WebRequest: Muestra una página web.

```
Invoke-WebRequest -Uri "http://www.google.com"
```

- Measure-Object: Este comando cuenta los archivos y carpetas del directorio actual.

```
Get-ChildItem | Measure-Object
```

- Resolve-Path: Muestra la ruta actual.

```
PS C:\> Resolve-Path ~
```

```
Path
```

```
-----  
C:\Users\User
```

- Resume-Job: Reanuda un trabajo por nombre.

```
PS C:\> Resume-Job -Name WorkflowJob, InventoryWorkflow, WFTest*
```

- Set-Variable: Establece una variable y obtiene su valor.

```
Set-Variable -Name "desc" -Value "Una descripción"
```

```
Get-Variable -Name "desc"
```

Name	Value
-----	-----
desc	Una descripción

- Show-Command: Abre un cmdlet en una ventana de comandos.

```
Show-Command -Name "Invoke-Command"
```





Invoke-Command

Parámetros para "Invoke-Command":

FilePathComputerName	FilePathContainerId	
FilePathRunspace	FilePathUri	FilePathVMId
FilePathVMName	Session	Uri
	VMId	VMName
InProcess	ComputerName	ContainerId

ScriptBlock: \*

ArgumentList:

InputObject:

☐ NoNewScope

Parámetros comunes

Ejecutar Copiar Cancelar

- Sort-Object: Ordena el directorio actual por nombre.

```
PS> Get-ChildItem -Path . | Sort-Object
```

Directory: C:\Users\User

Mode	LastWriteTime	Length	Name
----	-----	-----	-----
-a----	2/13/2019 08:55	26	anotherfile.txt

-a----	2/13/2019	13:26	20 Bfile.txt
-a----	2/12/2019	15:40	118014 Command.txt
-a----	2/1/2019	08:43	183 CreateTestFile.ps1
d-----	2/25/2019	18:25	Files
d-----	2/25/2019	18:24	Logs
-ar---	2/12/2019	14:31	27 ReadOnlyFile.txt
-a----	2/12/2019	16:24	23 Zsystemlog.log

- Start-Service: Inicia un servicio detenido.

```
Start-Service -Name "eventlog"
```

- Start-Process: Inicia un proceso que use valores por defecto.

```
Start-Process -FilePath "sort.exe"
```