



Certified Tech Developer

The Ultimate Degree

Infraestrutura I

Glossário de Comandos

A seguir, veremos alguns dos **cmdlets** mais úteis e comuns que usaremos no Windows PowerShell.

25 comandos básicos de Windows no PowerShell:

Cmdlet	Descrição.
Set-Location	Define o local de trabalho atual para um local específico.
Get-Content	Obtém o conteúdo do elemento no local especificado.
Add-Content	Adiciona conteúdo aos elementos especificados, por exemplo, adicionando palavras a um arquivo.
Set-Content	Escreve ou substitui o conteúdo de um elemento por conteúdo novo.
Copy-Item	Copia um item de um local para outro.
Remove-Item	Exclui os elementos especificados.
Move-Item	Move um item de um local para outro.
Set-Item	Altera o valor de um elemento para o valor especificado no comando.
New-Item	Cria um item novo.
Start-Job	Inicia um trabalho em segundo plano do Windows PowerShell.
Compare-Object	Compara dois conjuntos de objetos.
Group-Object	Agrupar objetos que contêm o mesmo valor nas propriedades especificadas.

Invoke-WebRequest	Obtém o conteúdo de uma página da Web na Internet.
Measure-Object	Calcula os valores de propriedade de certos tipos de objetos. Você pode realizar três tipos de medições dependendo dos parâmetros do comando.
Resolve-Path	Resolve os caracteres curinga em um caminho e exibe o conteúdo do caminho.
Resume-Job	Reinicia um trabalho suspenso.
Set-Variable	Define o valor de uma variável. Crie a variável se não existir com o nome solicitado.
Show-Command	Exibe comandos do Windows PowerShell em uma janela de comando gráfica.
Sort-Object	Classifica objetos por valores de propriedade.
Start-Service	Inicia um ou mais serviços parados.
Start-Process	Inicia um ou mais processos no computador local.
Suspend-Job	Interrompe temporariamente um trabalho.
Wait-Job	Aguarda os trabalhos em segundo plano do Windows PowerShell completar antes de exibir o prompt de comando.
Where-Object	Seleciona os objetos do conjunto de objetos passados a ele.
Write-Output	Envia o objeto especificado para o próximo comando no pipeline. Se for o último comando no pipeline, o objeto será exibido no console.

Exemplos de uso

- Set-Location: Define a localização atual.

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

- Get-Content: Obtem o conteúdo de um arquivo de texto.

```
Get-Content -Path .\LineNumbers.txt
```

```
Esta é a linha 1
```

```
Esta é a linha 2
```

- Add-Content: Adiciona conteúdo a um arquivo de texto.

```
Add-Content -Path .\LineNumbers.txt -Value 'Fim do arquivo'
```

```
Esta é a linha 1
```

```
Esta é a linha 2
```

```
Fim do arquivo
```

- Set-Content: Modifica uma linha de um arquivo de texto.

```
(Get-Content -Path .\LineNumbers.txt) |
```

```
ForEach-Object {$_ -Replace 'Fim do arquivo', 'Esta é a linha 3'} |
```

```
Set-Content -Path .\LineNumbers.txt
```

```
Get-Content -Path .\LineNumbers.txt
```

```
Esta é a linha 1
```

```
Esta é a linha 2
```

```
Esta é a linha 3
```



- Copy-Item: Copia um arquivo para um diretório específico.

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

- Remove-Item: Remove da pasta atual todos os arquivos que possuem uma extensão de nome de arquivo .doc com exceção daqueles que possuem um *1* em seu nome.

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

- Move-Item: Move um arquivo para outro diretório e renomeie-o.

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

- Set-Item: Cria um alias chamado np para Windows Notepad.

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

- New-Item: Cria um arquivo no diretório atual.

```
New-Item -Path . -Name "arquivo.txt" -ItemType "file" -Value "texto de  
teste"
```

- Start-Job: Executa um script como um processo em segundo plano.

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

- Compare-Object: Compara o conteúdo de dois arquivos.

Este exemplo compara o conteúdo de dois arquivos, o exemplo usa os dois arquivos de texto a seguir, cada um com um valor em uma linha diferente.

- vegetais.txt contém os valores: batata, abobora, tomate.
- frutas.txt contém os valores: banana, tomate, laranja.

A saída mostra apenas as linhas que são diferentes entre os arquivos. em vegetais.txt use o objeto de referência (<=) e nas frutas.txt use o objeto de diferença (=>). As linhas encontradas em ambos os arquivos não são exibidas.

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

```
InputObject SideIndicator
```

```
-----
```

```
banana      =>
```

```
laranja     =>
```

```
batata      <=
```

```
abobora     <=
```

- Group-Object: Agrupa os arquivos por sua extensão.

```
$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: Mostra uma página da web.

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

- Measure-Object: Este comando conta os arquivos e pastas no diretório atual.

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

- Resolve-Path: Mostra a rota atual.

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

```
Path
```

```
----
```

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

- Resume-Job: Retoma um trabalho pelo nome.

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

- Set-Variable: Define uma variável e obtém seu valor.

```
Set-Variable -Name "desc" -Value "Uma descrição"
```

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

```
Name
```

```
Value
```

```
----
```

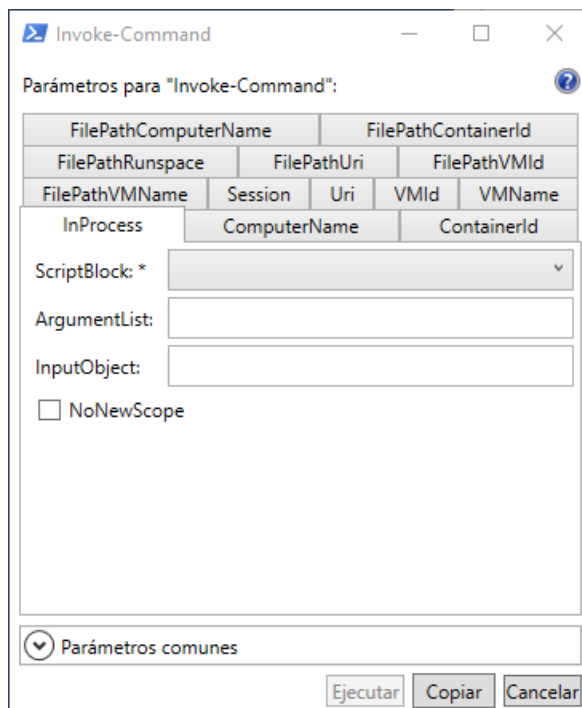
```
-----
```

```
desc
```

```
Uma descrição
```

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

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



Invoke-Command

Parámetros para "Invoke-Command":

FilePathComputerName	FilePathContainerId
FilePathRunspace	FilePathUri
FilePathVMName	Session
Uri	VMId
VMName	ContainerId

InProcess

ScriptBlock: *

ArgumentList:

InputObject:

☐ NoNewScope

Parámetros comunes

Ejecutar Copiar Cancelar

- Sort-Object: Classifica o diretório atual por nome.

```
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 um serviço interrompido.

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

- Start-Process: Inicia um processo que usa valores padrão.

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