

# Windows PowerShell

Es una consola de administración de sistemas incluida en todos los sistemas Windows y que puede ser instalada en sistemas Linux y MacOS.

A diferencia de otras consolas esta se comunica con objetos y no con textos. Es decir que el resultado de cada comando puede devolver valores en forma de objetos complejos.

Utiliza un lenguaje para comunicarse con la misma que conjuga verbos con nombres para determinar las acciones que queremos indicarle al sistema.

## Verbos y Nombres

Ejemplos de verbos serían: Get, Set, New, etc...

Ejemplos de nombres: Localuser, Localgroup, Partition, Disk, ComputerInfo, etc...

Cada verbo tiene la posibilidad de utilizar muchos de esos nombres para realizar las operaciones más habituales que necesitamos para administrar el sistema. Por ejemplo "Get" se utilizará para obtener información, "Set" para configurar algún valor, "New" para crear elementos de algún tipo, etc...

Ejemplos:

- **Get-Host:** versión de PowerShell
- **Get-Localuser:** Nos informa del usuario actual.
- **Get-Command:** Muestra los comandos disponibles en el sistema
- **Get-ComputerInfo:** Las características de nuestro sistema.
- **Set-Location:** cambia la ubicación en el árbol de directorios.

## Ayuda del sistema

Como la cantidad de comandos posible en el sistema es muy extensa y cada comando contiene a su vez muchas posibilidades de utilización, se necesita un buen sistema de ayuda que nos permita obtener información detallada y ejemplos del uso de cada uno.

La ayuda de PowerShell cambia constantemente ofreciendo nuevos ejemplos y actualizando la descripción de los comandos. Para poder actualizar nuestro sistema de ayuda debemos ejecutar el comando (en modo administrador):

### **Update-Help**

Si queremos obtener ayuda de un comando en particular debemos ejecutar el comando:

**Get-Help "nombre del comando"**

Si queremos obtener ejemplos para ese comando debemos agregar el indicador "-examples":

**Get-Help "nombre del comando" -examples**

Por ejemplo, podemos ejecutar el siguiente comando:

**Get-Help Get-ComputerInfo**

Y el resultado sería el siguiente:

#### NOMBRE

Get-ComputerInfo

#### SINOPSIS

Gets a consolidated object of system and operating system properties.

#### SINTAXIS

Get-ComputerInfo [[-Property] <String[]>] [<CommonParameters>]

#### DESCRIPCIÓN

The Get-ComputerInfo cmdlet gets a consolidated object of system and operating system properties.

#### VÍNCULOS RELACIONADOS

Online Version: <http://go.microsoft.com/fwlink/?LinkId=822226>

#### NOTAS

Para ver los ejemplos, escriba: "get-help Get-ComputerInfo -examples".

Para obtener más información, escriba: "get-help Get-ComputerInfo -detailed".

Para obtener información técnica, escriba: "get-help Get-ComputerInfo -full".

Para obtener ayuda disponible en línea, escriba: "get-help Get-ComputerInfo -online"

En el apartado NOTAS, nos indican otros posibles indicadores que podemos utilizar para ampliar la ayuda de este comando. Entre estos destaca “-examples” para obtener ejemplos, “-detail” para una ayuda más detallada y “-online” que nos redirige a la web de ayuda de Microsoft con una documentación mucho más extensa y organizada.

## Módulos

Cada módulo que añadamos a PowerShell contiene un grupo nuevo de comandos que podemos utilizar

*Get-Module -> muestra todos los módulos cargados en memoria*

Get-command -module Microsoft.PowerShell.Management -> muestra los comandos de un módulo

Get-Module -ListAvailable -> lista módulos disponibles para importar

Import-Module BitLocker -Verbose -> cargamos el módulo en memoria

Remove-Module BitLocker -> lo quitamos de memoria

## Atajos

Tabulador: El uso del tabulador nos permite autocompletado

Flechas: Permite reutilizar comandos

Get-history: nos muestra comandos del historial (también *h*)

Invoke-history *Num*

*r Num*

*Control+R*: permite teclear texto y buscar en el historial

Clear-history -> Borra el historial

## Alias

Los alias son apodos más rápidos de usar para los comandos

Get-Alias -> Muestra lista de alias

Get-alias -definition *Comando* -> Muestra los alias de un comando

## Comandos para archivos y carpetas

Get-Location (pwd) : Ruta o path

Set-Location (cd) : Para desplazarse.

Get-ChildItem (ls) : Muestra el contenido.

[-attributes hidden]

New-Item (ni/md) : Para crear archivos y carpetas.

New-item -itemType Directory nombre\_dir

New-item nombre\_fichero

New-item nombre1, nombre2, ....

Remove-Item (rm) : Para Borrar archivos y carpetas.

Remove-item nombre\_directorio -recurse (borrado recursivo)

Move-Item (mv) : Para mover archivos y carpetas.

Rename-Item (ren) : Para renombrar archivos y carpetas.

Copy-Item (cp) : Para copiar archivos y carpetas. [-recurse]

Get-content (cat) :

## Tuberías y Redireccionamiento

Get-command | measure-object

Get-ChildItem -Recurse | Where-Object {\$\_.Length-gt 100Mb} Búsqueda de ficheros con más de 100Mb

Get-ChildItem -Recurse | Where-Object {\$\_.Length-gt 100Mb} | Sort-Object -Descending -Property Length Ordenando por longitud

Get-netTCPConnection | ft -a formatea como tabla para ver todos los datos

Get-netTCPConnection | Where-Object{\$\_.State -eq 'Established'} | ft -a Con estado 'Established'

> y >> Igual que en cmd

## Obtener todos los campos de un resultado

Mediante el comando Get-Member podemos obtener todos los campos que devuelve un resultado de cualquier comando:

Get-ChildItem | Get-Member