

Introducción a PowerShell

En las versiones basadas en MS-DOS, recordemos que la última versión fue Windows Me, permitían ejecutar comandos mediante una línea de comandos que se llamaba «Command.com», al estar basado en MS-DOS los comandos que se ejecutaban eran propios de ese sistema operativo.

En las versiones basadas en NT también se podía ejecutar comandos de MS-DOS aunque el corazón del sistema operativo no lo fuera. En estas versiones el programa para ejecutar comandos se llamaba Cmd.exe. Los comandos que se ejecutaban en las versiones basadas en NT eran los mismos que en MS-DOS y además los propios de Windows como por ejemplo los comandos: net (administrar recursos de red), netstat (estadísticas de la red), ping (probar una conexión de red), subst (asocia una ruta con una letra de unidad), etc.

En todas las versiones de Windows existe un programa que se llama «Símbolo de sistema» en donde se pueden ejecutar comandos, se accede pulsando en el «Botón de Inicio» – «Todos los programas» – «Accesorios» – «Símbolo de sistema».

Desde el año **2006** Microsoft dispone de una línea de comandos mejorada que se denomina **PowerShell** (aunque los **orígenes** de PowerShell datan de años anteriores y el nombre del proyecto se conocía como **MONAD**), es más **potente** y **rica** que la consola de **MS-DOS**.

PowerShell se ha creado sobre **.NET Framework, Common Language Runtime (CLR) y .NET Framework**, y **acepta y devuelve objetos de .NET**.

Algunas características de PowerShell son:

- Windows 7 incorpora PowerShell, también viene instalado en Windows Server 2008 R2 y se puede instalar en versiones anteriores de Windows como Windows XP, Vista,

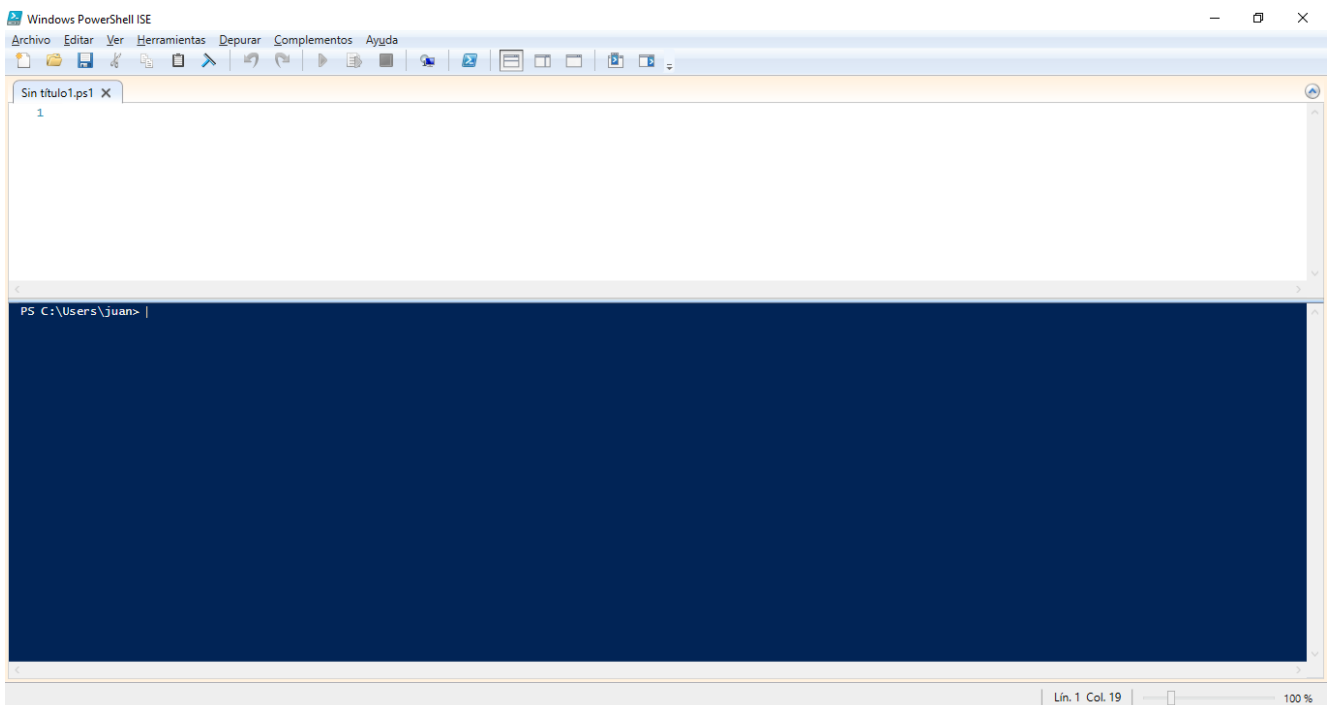
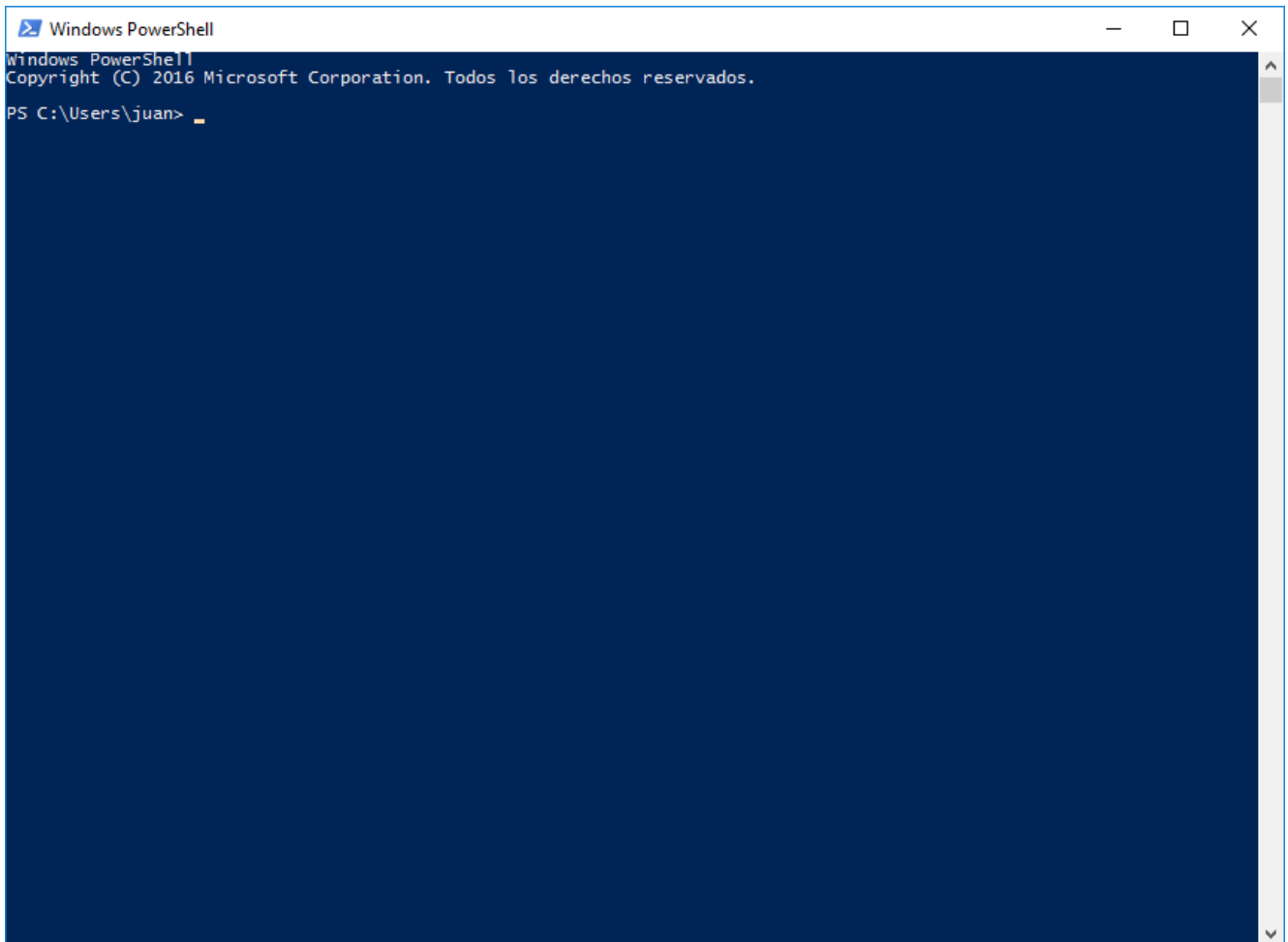
Server 2003, Server 2008.

- Está disponible para x86 y x86-64.
- Los comandos u órdenes en PowerShell reciben el nombre de «cmdlets».
- Es una consola orientada a objetos.
- PowerShell está basado en .NET Framework de Microsoft. .NET Framework es un conjunto de módulos de software, bibliotecas, librerías, programas, etc., que ayudan al desarrollo de otros programas.
- Permite acceso a almacenes de datos como por ejemplo el Registro de Windows.
- Se pueden ejecutar scripts creados en Windows Script Host. Windows Script Host ofrece la posibilidad de crear archivos por lotes complejos y potentes.
- Incorpora las clases del Instrumental de administración de Windows (Windows Management Instrumentation, WMI). WMI es un conjunto de funciones y procedimientos del sistema operativo Windows para controlar, monitorear y administrar los equipos en una red.
- Ofrece un completo sistema de ayuda.
- Permite utilizar la tecla tabulador para completar el nombre de un comando, fichero, carpeta, etc.
- Se pueden crear y utilizar cmdlets nuevos, funciones y scripts.

PowerShell es una línea de comandos con tecnología de **scripting** basada en tareas que proporciona a los administradores de **tecnologías de la información (TI)** un control integral y la posibilidad de **automatizar las tareas** de administración del sistema.

El programa que sirve para ejecutar PowerShell se llama powershell.exe y la versión ISE (el entorno de script integrado) se llama powershell_ise.exe.

Para acceder a PowerShell hay que pulsar «Inicio» – «Todos los programas» – «Accesorios» – «Windows PowerShell».



Los comandos (**cmdlets**, se pronuncia command-let) de PowerShell son sencillos de recordar usan el sistema **verbo-nombre** para llamar a los comandos, los **verbos** y los **nombres están en inglés**. Algunos ejemplos de cmdlets son:

[crayon-6a9d575144029424505623/]

```
Sin título1.ps1 X
1 Get-Process

PS C:\Users\juan> Get-Process

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
275 17 3636 3864 1,77 5936 1 AsusTPCenter
69 7 1132 348 0,02 5956 1 AsusTPHelper
210 12 2260 652 0,03 5400 1 AsusTPLoader
274 18 5764 21620 0,14 6336 1 backgroundTaskHost
406 52 164976 185040 38,36 4200 1 chrome
1363 75 89056 129332 132,89 5280 1 chrome
167 11 2000 8632 0,02 5308 1 chrome
125 11 1916 9036 0,03 5336 1 chrome
398 29 136428 146404 86,92 5480 1 chrome
354 33 32920 61988 5,27 7876 1 chrome
325 13 1384 3728 608 0 csrss
309 13 1944 4388 700 1 csrss
211 12 3352 13388 2124 0 dasHost
169 19 4032 10328 76 1 dllhost
307 21 31920 43104 604 1 dwm
83 6 1244 4440 0,02 3620 1 esif_assist_64
131 8 1748 6652 2364 0 esif_uf
2130 114 47300 100160 14,70 3828 1 explorer
0 0 0 4 0 0 idle
162 9 1840 8556 1376 0 igfxCUIService
175 13 3584 12160 0,05 3984 1 igfxEM
112 9 2172 8692 0,02 4028 1 igfxHK
152 11 3180 10820 0,27 4076 1 igfxTray
104 7 1292 6480 2372 0 KinectManagementService
109 7 1800 6092 2356 0 KinectMonitor
428 24 166808 81228 3,59 3672 1 KinectService
991 21 5080 13736 832 0 lsass
0 0 0 8 2672 0 Memory Compression
262 13 3256 10548 6636 0 MpCmdRun
309 14 3496 12544 0,17 5568 1 MSAScuiL
529 64 120448 76996 2752 0 MsMpEng
268 11 6068 3080 3564 0 NISrv
717 91 133228 156704 3720 1 powershell_ise
254 14 5472 648 0,05 5172 1 RAVB064
308 14 3768 900 0,20 5164 1 RAVBp164
666 22 11126 36556 6,44 3616 1 RuntimeBroker
```

Los cmdlets usan parámetros que son propios, esto quiere decir que los parámetros de un Cmdlet pueden no ser los mismos que los de otro, aunque en general son los mismos. Algunos ejemplos de parámetros son: -Name, -Path, -ComputerName, etc.

PowerShell tiene la posibilidad de crear «alias», un alias permite renombrar o llamar a los comandos de distintas formas. Esto es muy útil porque se pueden crear alias con nombres de parámetros que utilicemos en otros sistemas operativos o que nos inventemos nosotros mismos, por ejemplo podemos utilizar el alias «ps» para listar los procesos, este comando es propio de Linux, también podemos crear un alias inventado por nosotros que sea «Dime-Procesos».