
AGRADECIMIENTOS

Sé que este es un libro de temática técnica y que yo soy su humilde redactor. Pero también sé que como persona, profesional y eslabón central de una cadena, mi trabajo hace que el conocimiento compartido por otras personas enriquezca a su vez a quien me lee.

Yo soy miembro del Estado español, que pertenece a la Unión Europea, algo que en términos de geografía nos limita a un territorio, pero que no es más que una parte del entramado de comunicaciones transoceánicas global y que a todos los territorios une en mayor o menor medida.

Esta interconexión, llevada a cabo por los sistemas de comunicación de fibra global, me hace pensar en lo relativo de la territorialidad, e igualmente me obliga a escribir estos agradecimientos técnicos y humanos.

La tecnología no existiría ni avanzaría sin el factor humano que la crea. ¿Por qué entonces estamos observando tanta carencia de humanidad y egoísmo en el entorno que nos rodea, en nuestra Europa?

Como empecé diciendo, no quiero perder el norte en la redacción, pero tampoco la humanidad.

Gracias, por tanto, a quien me enseñó que todas las personas son necesarias para que nuestro sistema tecnológico pueda sostenerse y mejorar. Gracias a las ONG que no permiten que el futuro de mi profesión avance con la pérdida de posibles profesionales que podrían aportar grandes mejoras.

En definitiva, como dijo Albert Einstein, alemán judío, emigrante/refugiado y sobre todo científico: “El espíritu humano debe prevalecer sobre la tecnología”.

INTRODUCCIÓN

Esta obra lo que ha pretendido conseguir desde el comienzo de su desarrollo, y lo que busca tras la lectura del lector, es que se pueda trabajar con Microsoft Windows PowerShell de una manera inmediata e independientemente de la versión que se posea de Microsoft Windows. El desarrollo, por tanto, ha tenido un enfoque práctico con referencias a la realidad teórica de este modelo de programación, intentando de esta manera que el contenido resulte lo más útil y funcional posible.

Cierto es que Microsoft Windows PowerShell no es solo lo que aquí se muestra, pero la extensión de la obra y el planteamiento de desarrollo anteriormente expuesto nos obliga a abordar aspectos concretos del mismo.

Se espera de esta manera que el lector encuentre en la obra un modelo ameno de exposición y que su utilidad coincida con lo esperado.

A lo largo de la obra encontrará anotaciones, ejemplos prácticos o pequeños trucos que pretenderán aclarar y hacer el proceso de lectura más fácil.

Cada uno de los iconos utilizados a lo largo de la obra que aportan información extra son:



NOTA

Información extra de carácter general.

HISTORIA, EVOLUCIÓN Y ACTUALIDAD

Microsoft Windows ha ido cambiando y buscando mantener su posición dentro de los usuarios de informática independientemente del nivel profesional en que se ubiquen. Esto ha hecho que se aventure en cambios estéticos y funcionales, así como estructurales. Quizás la parte estructural en la que han fortalecido la seguridad de sus sistemas sea una de las principales áreas donde han trabajado.

Dentro de este apartado estructural es donde yo incluiría los cambios que se han aportado con la inclusión de la consola de Microsoft Windows PowerShell, una consola que no sería extraño pensar que terminará sustituyendo totalmente a la hasta ahora emblemática pero insegura **cmd** (*comand Shell*).

Antes de empezar a trabajar con esta consola y todas las posibilidades que nos aporta, veamos cuál ha sido su corta historia, los requisitos y sus versiones.



Figura 1.1. Logotipo Microsoft Windows PowerShell

1.1 ¿QUÉ ES MICROSOFT WINDOWS POWERSHELL?

Cuando hablamos de Microsoft Windows PowerShell estamos refiriéndonos a la actual consola de comandos o CLI (*Command-Line Interface*). Esta consola nos aporta la posibilidad de trabajar directamente sobre ella, así como poder automatizar tareas gracias a la creación de pequeños programas o *scripts*.

Microsoft Windows PowerShell es una consola con mucha más funcionalidad e interacción de la que hemos podido gozar en su antecesora. De manera que la hace una consola imprescindible para aquellos administradores que quieran trabajar con Microsoft Windows Server.



NOTA

Hasta 2006, Microsoft Windows PowerShell era conocida y publicitada por parte de Microsoft con el nombre de **Monad**. Podemos ver los artículos iniciales en la página web de la propia Microsoft. <https://technet.microsoft.com/en-us/magazine/2005.11.scripting.aspx>

Installing Monad

[**Editor's Update - 5/22/2006:** Monad has been renamed Windows PowerShell. Click [here](#) for more information, and to download a copy of Windows PowerShell.

]

Monad is currently in beta testing. Microsoft has released Monad both to a technical beta program and to the wider public as a part of the WinFx™ SDK beta. To install the Monad

1.2 NOVEDADES FRENTE AL CMD CLÁSICO

La consola **cmd** de Microsoft Windows ha estado presente en los sistemas operativos de Microsoft desde sus versiones DOS hasta Windows 7. En las versiones posteriores de Microsoft Windows 7, es decir, desde la 8 a la actual 10 y en las actuales Microsoft Server, no es que hayan eliminado la **cmd** clásica en pro de esta nueva conocida como Microsoft Windows PowerShell, sino que conviven ambas. De manera que si queremos usar Microsoft Windows PowerShell la tendremos que ejecutar de manera independiente y consciente.

Microsoft Windows PowerShell se puede definir como un programa que permite a los usuarios privilegiados o administradores el acceso a los diferentes servicios vinculados al núcleo del sistema. El porqué del interés en el uso de esta consola es porque nos aportará un modelo de trabajo más rápido para el desarrollo de algunas tareas, gracias a la liberación de cargas, entre otras la gráfica. Aunque en el antiguo **cmd** también se podían realizar parte de estas tareas, Microsoft Windows PowerShell ha ampliado con creces el abanico.

En cuanto al desarrollo de *scripts* o programas administrativos, Microsoft Windows PowerShell ha dotado a su sistema de *scripting* del concepto de seguridad del que **cmd** carecía. De esta manera, si un usuario malintencionado generase un

código malicioso no se podría ejecutar de manera automática en nuestra maquina sin pasar por unos parámetros de seguridad. Algo que en la antigua **cmd** nos dejaba “vendidos” en multitud de casos.

A este aumento de la seguridad, se le suma la interacción directa con el *framework* .NET.

**NOTA**

Aunque han cambiado multitud de cosas, no hay que asustarse, pues el personal de desarrollo de Microsoft ha tenido en cuenta la migración de usuarios y ha facilitado su uso gracias a los **alias**.

1.3 REQUISITOS

Los requisitos son pocos, por no decir único. Para su correcto funcionamiento se requiere tener instalado el *framework* .NET versión 3.0.

**NOTA**

Aunque con la versión de *framework* .NET versión 2.0 podría valer, se recomienda tener instalada la versión 3.0 para el correcto funcionamiento de servicios complementarios, como es el Microsoft Windows PowerShell ISE.

1.4 VERSIONES

Microsoft Windows PowerShell ha pasado por 5 versiones, siendo la actual durante el desarrollo de esta obra la versión 5.

- Versión 1.0, convivió con Microsoft Windows XP y Server 2013. Su descarga independiente era necesaria, al no ser aportado con los sistemas operativos en el proceso de instalación.
- Versión 2.0, incluyó Microsoft Windows PowerShell ISE, un entorno de desarrollo que facilitaba la programación de *scripts*. Se incluyó en Microsoft Windows 7 y Server 2008 R2.
- Versión 3.0, mejoró el flujo de trabajo y amplió el número de comandos útiles conocido como **cmdlets**.

- Versión 4.0, incluye soporte para flujo de trabajo y la depuración de *scripts* remoto, mejora el soporte para la descarga de ayuda de manera que sea actualizable y mejora el rendimiento.
- Versión 5.0, nuevos cmdlets, entre los que incluiremos algunos que nos permitirán trabajar con ficheros .zip, o manejador de paquetes que permitirá instalación o desinstalación de programas remotos a través de la consola. Coloreado de sintaxis. La *preview* fue liberada en abril de 2015.



NOTA

Si se está interesado en ver las diferencias existentes en relación a la versión de soporte de cada una de las versiones, se recomienda acudir a las notas técnicas de Microsoft en la web <https://technet.microsoft.com/en-us/library/dn997309%28v=sc.16%29.aspx>

- TechNet Library
- System Center
- System Center 2016 Technical Preview
- System Requirements
- Hardware Recommendations
- Server Operating Systems
- Client Operating Systems
- SQL Server Versions
- Console Compatibility
- PowerShell Version Support
- System Center Agents
- .NET Version Support

PowerShell Version Support

Published: May 4, 2015
 updated: April 27, 2016
 Applies To: System Center Technical Preview

Use this information to evaluate if your Windows PowerShell environment is ready to support the installation of or upgrade to System Center 2016 Technical Preview. Use this information here whether you are deploying one or multiple components.

• PowerShell support

System Center Component	Windows PowerShell 2.0	Windows PowerShell 3.0	Windows PowerShell 4.0	Windows PowerShell 5.0
DPM Central Console	*	*	*	*
DPM Remote Administration console	*	*	*	*
DPM Server	*	*	*	*
Operations Manager Management Server			*	*
Operations Manager Data Warehouse Server			*	*
Operations Manager Gateway Server			*	*
Operations Manager Windows Agent	*	*	*	*
Service Manager Console			*	*
Service Manager Management Server			*	*
Service Manager Data Warehouse Management Server			*	*
Service Provider Foundation API Web Server			*	*
Virtual Machine Manager Management Server			*	*
Virtual Machine Manager Console			*	*

1.5 EJERCICIOS

1. Investigar lo que en un principio fue el proyecto Monad y que ha quedado de él finalmente en Microsoft Windows PowerShell. Para ello hacer uso del artículo original publicado en las notas técnicas de Microsoft <https://technet.microsoft.com/en-us/magazine/2005.11.scripting.aspx>.
2. Buscar en la Wikipedia la palabra **Batch** con la intención de saber qué era y cómo se usaba el clásico **CMD**.
3. Ver qué es el *framework* .NET. Hacer uso de Internet para ello.