

# Histórico de los lenguajes de script

Al comienzo de la historia de la informática, existía Unix... este sistema operativo salió a la luz en 1968, y con él surgieron los primeros entornos de ejecución de scripts llamados « Shells ». He aquí alguno de ellos, por citar los más conocidos (en orden cronológico): Bourne shell (sh), C shell (csh), Korn shell (ksh), bash shell (el shell bash es un proyecto GNU iniciado por la Free Software Foundation). Se denominan « script Shell » los scripts desarrollados por estos entornos. Todavía son muy utilizados para la administración de los sistemas Unix.

Durante ese tiempo, Microsoft desarrollaba las primeras versiones de DOS en Bellevue en los suburbios de Seattle (la primera versión se llamaba Q-DOS para *Quick and Dirty Operating System*, sistema operativo realizado a prisa y corriendo). Es en 1981 que el MS-DOS hizo su aparición en su versión 1.0 con los primeros scripts batch (archivos de extensión .bat). Estos últimos eran (y siguen siendo) muy limitados en terminos de funcionalidades, pero todavía son muy utilizados por los administradores Windows para la realización de tareas sencillas.

1987 es el año del lanzamiento del lenguaje PERL (*Practical Extraction and Report Language*), desarrollado por Larry Wall, en su versión 1.0, para las plataformas Unix. Este lenguaje - cuyas funcionalidades se basaron en los scripts shell y el lenguaje C - tiene por objetivo la manipulación de archivos, de datos y de procesos. Fué necesario esperar diez años más para ver aparecer PERL en el mundo Windows, en 1997 en el kit técnico de recursos de Windows NT 4.0.

Existen otros muchos lenguajes de scripts orientados al sistema, que no detallaremos, ya que no es el objetivo de esta obra, pero que merecen al menos ser citados, es el caso de: Rexx, Python, S-Lang, Tcl/tk, Ruby, Rebol, etc.

Otros dos lenguajes de script muy extendidos en la Web són Javascript y VBScript. El primero fue desarrollado por Netscape y hecho público en 1995 (en Navigator 2.0) con el fin de aportar a los sitios de Internet más dinamismo y vida, así como aportar mayor interactividad a las páginas HTML (*HyperText Markup Language*). La respuesta de Microsoft no se hizo esperar con la aparición del lenguaje JScript (en Internet Explorer 3.0), lenguaje muy similar al de Netscape. Para poner a todo el mundo de acuerdo, el ECMA normalizó estos lenguajes en 1997 tomando lo mejor de cada uno para formar el ECMAScript (ISO/IEC 16262) (el ECMA es un organismo internacional de normalización de sistemas de información y de comunicación). Fué también en Internet Explorer 3 que VBScript hizo su aparición. VBScript recoge la totalidad de las funcionalidades de Javascript pero con una sintaxis próxima a Basic, mientras que Javascript se parece más a Java o a C.

Volviendo al desarrollo de los scripts de sistema, antes de la llegada del kit técnico de recursos de NT 4.0, los administradores de sistemas Windows no tenían otra elección que utilizar los scripts batch MS-DOS. Estos últimos eran sencillos cuando se trataba de realizar tareas triviales (como por ejemplo montar una unidad de red), pero podían resultar muy laboriosos cuando se deseaba analizar un fichero, o realizar un simple bucle. Afortunadamente Microsoft tuvo la gran idea de poner en el kit de recursos, además de Perl, un intérprete de script KiXtart. KiXtart extendió de forma significativa los scripts batch aportando numerosas funcionalidades de red, así como un poderoso lenguaje de script realmente pensado para la administración del sistema. En resumen, KiXtart es una alternativa válida a los viejos scripts batch. Sólo una pequeña objeción, todo y que Kix fué desarrollado por miembros de Microsoft, no está oficialmente soportado por Microsoft.

Es en este momento que aparece discretamente con Windows 98 y con el option Pack de NT4.0, una fabulosa herramienta de administración llamada WSH (*Windows Script Host*). Esta última ofrece un poderoso entorno de ejecución de scripts VBScript, pero fue poco conocida en sus inicios. Fue así que a espaldas de los usuarios, numerosos virus utilizaron las nuevas funcionalidades (y fallos de seguridad) ofertadas por WSH, dando a conocer de este modo sus capacidades...

No obstante en la actualidad WSH/VBScript es la pareja más extendida para realizar tareas de administración de sistema en el mundo Windows, pero esto puede fácilmente cambiar próximamente...

## 1. ¿Y PowerShell entre todo esto?

Microsoft tomó conciencia, alrededor de 2004-2005, de los límites de la interfaz gráfica y de las dificultades experimentadas por los usuarios para la realización de scripts. Entonces, la empresa de Redmond cambió radicalmente su visión de las cosas y dio un giro de ciento ochenta grados anunciando el nacimiento de Monad. Monad era el nombre del código de PowerShell en sus versiones preliminares; la versión final de PowerShell en su versión 1.0 se publicó a finales de 2006. En cuanto a la versión 2.0, la última en aparecer, se encuentra integrada en Windows Server 2008 R2 y

Windows 7. La versión descargable estuvo disponible a finales de octubre de 2009.

La nueva estrategia de Microsoft para la administración de sistemas de sus productos está hoy en día claramente orientada a la línea de comandos. Si bien se mantendrán las interfaces gráficas de las herramientas de administración de sistema -que contribuyeron al éxito de Windows-, a partir de ahora estarán construidas sobre PowerShell. Por ejemplo, con Microsoft Exchange 2007 y ahora con Microsoft Exchange 2010, las acciones realizadas gráficamente ejecutan en realidad comandos PowerShell que se pueden recuperar para integrarlos en los scripts.

PowerShell está profundamente arraigado en Windows Server 2008 R2 y cada nuevo rol instalado lleva consigo nuevos comandos. Existen actualmente comandos para gestionar: Active Directory, las directivas de grupos, el clustering, AppLocker, las transferencias de archivos inteligentes (BITS), IIS, la gestión de roles y funcionalidades, etc.

Pero PowerShell se encuentra igualmente en el núcleo de numerosos productos Microsoft y no Microsoft como: Exchange Server 2007/2010, Virtual Machine Manager a partir de la versión 2007, SQL Server 2008, Microsoft Deployment Toolkit 2010, SharePoint 2010 e incluso VMware vSphere PowerCLI, Citrix XenApp, etc. Inclusive editores de programas informáticos diferentes de Microsoft desarrollan comandos PowerShell; esto demuestra el interés que suscita PowerShell...

PowerShell además aporta un intérprete de línea de comandos interactivo que faltaba en WSH y que estaba más que limitado con el intérprete de comandos CMD.exe. Cabe destacar en PowerShell la presencia de un juego de comandos coherente. Como cosa revolucionaria, PowerShell se basa en la tecnología .NET para dar una dimensión de objetos a los scripts y dar así acceso a la inmensa Biblioteca de clases del Framework .NET. Para concluir, la seguridad en PowerShell ha sido la preocupación constante de los miembros del equipo de desarrollo. De este modo descubrirá a lo largo de este libro que PowerShell es realmente una revolución por si mismo, y que modificará profundamente la forma de administrar los sistemas Windows de la próxima década.