

Diferencias entre CMD y PowerShell

Una comparación detallada de las herramientas de línea de comandos de Windows



CMD



PowerShell

Introducción a CMD y PowerShell

CMD (Símbolo del sistema)

Origen

Desarrollado en los **años 80** para MS-DOS

Lenguaje

Basado en **texto plano** y comandos batch

Propósito

Ejecutar comandos básicos del sistema

Limitaciones

Capacidades limitadas de scripting y automatización

PowerShell

Origen

Introducido en **2006** como sucesor de CMD

Lenguaje

Basado en **objetos** y cmdlets (verbo-nombre)

Propósito

Automatización avanzada y administración de sistemas

Ventajas

Potente motor de scripting y gestión de objetos

Diferencias principales entre CMD y PowerShell

Aspecto	CMD	PowerShell
<> Lenguaje	Comandos tradicionales	Cmdlets con formato verbo-nombre
{ } Estructura de datos	Basado en texto plano	Basado en objetos estructurados
Scripting	Lenguaje simple y limitado	Motor de scripting potente y flexible
⚙️ Casos de uso	Diagnósticos rápidos y tareas básicas	Automatización avanzada y administración de sistemas
? Ayuda integrada	Limitada	Get-Help con ejemplos y documentación detallada

Tabla de comandos de PowerShell

Get-Command

Lista todos los comandos disponibles

Get-Help

Muestra ayuda sobre comandos específicos

Get-Process

Muestra procesos en ejecución

Get-Service

Lista servicios del sistema

Stop-Process

Detiene procesos específicos

ConvertTo-HTML

Convierte resultados a formato HTML

Where-Object

Filtra resultados de comandos

Get-History

Muestra historial de comandos

Out-File

Guarda resultados en archivo

Copy-Item

Copia archivos y carpetas

Get-ChildItem

Lista contenido de directorios

Set-AuthenticodeSignature

Añade firma digital a scripts

Invoke-Item

Ejecuta acción predeterminada en archivos

Start-Job

Inicia tareas en segundo plano

Clear-Content

Elimina contenido de archivos

ForEach-Object

Ejecuta acción sobre cada objeto

Extracción de información del sistema con PowerShell

```
Product ID: 00378-00000-00000-AA739
Original Install Date: 5/6/2017, 6:51:03 PM
System Boot Time: 8/6/2017, 7:55:05 AM
System Manufacturer: Microsoft Corporation
System Model: Virtual Machine
System Type: x64-based PC
Processor(s): 1 Processor(s) Installed.
```

Información de escritorio

Obtiene datos sobre los escritorios del sistema

```
Get-CimInstance -ClassName Win32_Desktop
```

Información del BIOS

Muestra detalles del BIOS del sistema

```
Get-CimInstance -ClassName Win32_BIOS
```

Información del procesador

Obtiene datos del procesador del sistema

```
Get-CimInstance -ClassName Win32_Processor
```

Información del sistema

Muestra datos del equipo y fabricante

```
Get-CimInstance -ClassName Win32_ComputerSystem
```

Actualizaciones instaladas

Lista todas las revisiones del sistema

```
Get-CimInstance -ClassName Win32_QuickFixEngineering
```

Información de discos

Muestra espacio libre y total de discos

```
Get-CimInstance -ClassName Win32_LogicalDisk -Filter
"DriveType=3"
```

Servicios del sistema

Lista todos los servicios y su estado

```
Get-CimInstance -ClassName Win32_Service
```

Sistema operativo

Obtiene versión y detalles del SO

```
Get-CimInstance -ClassName Win32_OperatingSystem
```

Instalación de OpenSSH Server con PowerShell

```
Name : OpenSSH.Client~~~~0.0.1.0
State : Installed

Name : OpenSSH.Server~~~~0.0.1.0
State : NotPresent
```

🔍 Verificar disponibilidad

1

Comprueba si OpenSSH está disponible para instalar

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH\*'
```

⬇️ Instalar componentes

Instala cliente y servidor OpenSSH

2

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

▶️ Iniciar servicio

3

Inicia el servicio SSHD manualmente

```
Start-Service sshd
```

⚙️ Configurar inicio automático

4

Configura el servicio para que inicie automáticamente

```
Set-Service -Name sshd -StartupType 'Automatic'
```

📌 Nota importante

La instalación crea automáticamente una regla de firewall llamada **OpenSSH-Server-In-TCP** que permite conexiones SSH entrantes en el puerto 22.

Conclusiones: Ventajas de PowerShell sobre CMD



Mayor potencia

Capacidades avanzadas para **automatización** y administración de sistemas complejos



Manejo de objetos

Trabaja con **objetos estructurados** en lugar de texto plano, permitiendo manipulación más precisa



Scripting avanzado

Lenguaje de scripting **completo y robusto** con estructuras de control y funciones complejas



Mayor seguridad

Políticas de ejecución y **firmas digitales** para proteger contra scripts maliciosos



Conclusión

PowerShell representa la evolución natural de la línea de comandos en Windows, ofreciendo capacidades superiores para administración y automatización de sistemas modernos.