

## 4. Introducción a PowerShell

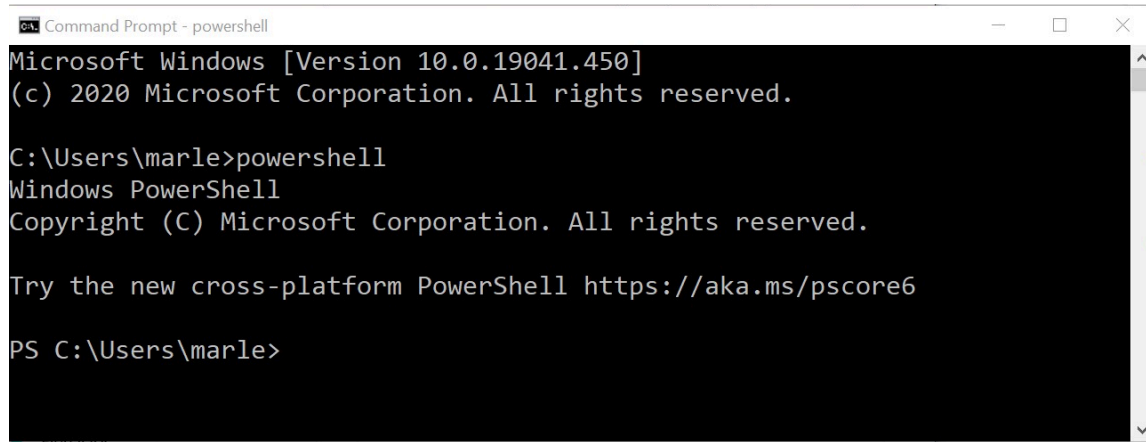
Monday, September 7, 2020 5:01 PM

### Introducción

- Microsoft PowerShell es un lenguaje de desarrollo, automatización universal y scripting. Se considera un puente para cerrar la brecha entre estas tareas.
- PowerShell es de código libre y multiplataforma.
- Contiene, dentro de sus comandos, los comandos de CMD.

### Iniciando en PowerShell

Existen múltiples opciones para comenzar a trabajar con PowerShell, una de ellas es inicializarlo desde el CMD.



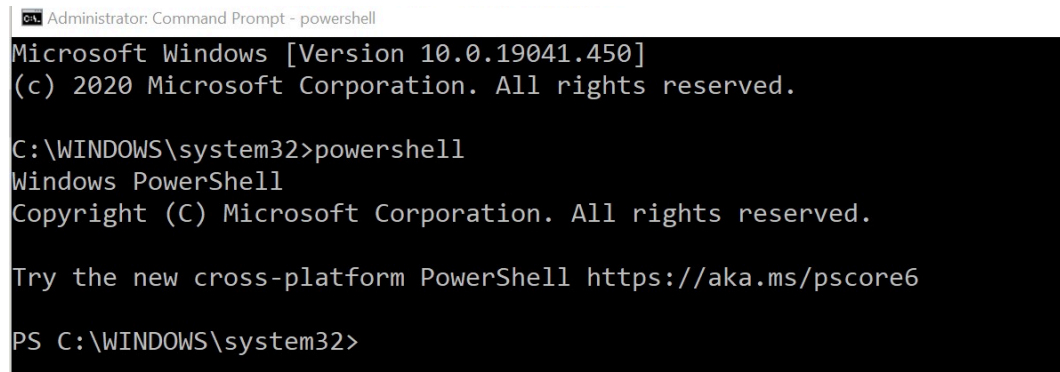
```
Command Prompt - powershell
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\marle>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\marle>
```

También podemos iniciar PowerShell desde el CMD ejecutado como administrador.



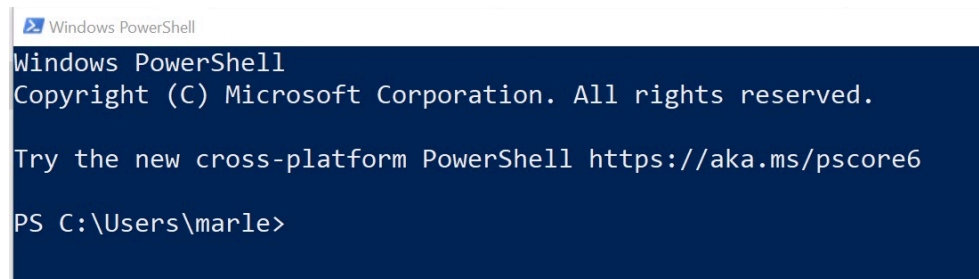
```
Administrator: Command Prompt - powershell
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32>
```

Podemos, además, buscar PowerShell en el menú de inicio de Windows y ejecutarlo.



```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\marle>
```

O, de igual forma, buscarlo en el menú de Windows pero ejecutarlo como Administrador.



```
Select Administrator: Windows PowerShell
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32>
```

¿Qué diferencias observas? En caso de que estés trabajando desde Linux o Mac, ¿cuál es la diferencia que observas al ejecutar como usuario normal a ejecutar como root?

### ? Comandos de cmd en PowerShell

Ya hemos mencionado que los comandos de cmd pueden ser usados en PowerShell, sin embargo, es importante aclarar que los "comandos" de cmd no son en realidad comandos, son alias o seudónimos que se traducen de comandos que tu conoces (los de cmd) a los que PowerShell conoce.

Ejemplo del uso de los comandos de CMD en PowerShell:

```
Administrator: Windows PowerShell
PS C:\WINDOWS\System32> cd ..
PS C:\WINDOWS> cd .\system32\
PS C:\WINDOWS\system32>
```

Para listar los comandos de CMD disponibles en PowerShell puedes usar el siguiente comando:

```
PS> Get-Alias
```

Un comando en PowerShell se puede definir como el término que usamos para llamar a expresiones ejecutables. El comando puede ser, prácticamente, sobre cualquier cosa, tanto un comando *legacy* de CMD, un comando propio de PowerShell, una llamada a un comando que tu hayas construido, la única restricción es que exista. En caso de que no exista, recibirás un "infame error rojo".

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> foo
foo : The term 'foo' is not recognized as the name of a cmdlet, function, script file,
or operable program. Check the spelling of the name, or if a path was included, verify
that the path is correct and try again.
At line:1 char:1
+ foo
+ ~~~
+ CategoryInfo          : ObjectNotFound: (foo:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\WINDOWS\system32>
```

Para conocer una lista de todos los comandos que PowerShell posee puedes utilizar la siguiente instrucción:

```
PS> Get-Command
```

Nota que el esquema general que siguen los comandos de PowerShell es Verbo-Sustantivo, lo cual es una característica representativa de PowerShell.

### Comandos de PowerShell

Los comandos de PowerShell pueden ser de diferentes tipos:

- **cmdlets.**- La mayoría de los comandos incorporados de Microsoft son de este tipo. Corresponden típicamente a comandos escritos en otros lenguajes de programación.
- **functions.**- Están escritas en PowerShell. Pueden ser funciones que tu hayas escrito.
- **alias.**- Como el ejemplo de los comandos de cmd
- **Script externos.**- Llamadas a scripts no nativos de PowerShell.

Al utilizar Get-Command podrás observar de que tipo es cada uno de los comandos. Puedes filtrar los resultados obtenidos de esta función a través de parámetros.



#### **parámetros**

Son los valores que tu ingresas a un comando para personalizar su comportamiento.

Intenta el siguiente código en tu consola de PS:

```
PS> Get-Command -Verb Get
```

#### **Referencias:**

- Bertram, Adam (2020) . *PowerShell for SysAdmins*. "no starch press"  
<https://nostarch.com/powershelladmins>