



**TSU EN TECNOLOGÍAS DE LA INFORMACIÓN ÁREA  
INFRAESTRUCTURA DE REDES DIGITALES.**

**Nombres:**

Cortes Hernández Nellie Montserrat.

Hernández Espinoza Thania.

**Asignatura:** Administración de servidores.

**Profesor:** Maribel Morales Guzmán.

**Fecha:** 28/03/2023

# Índice

|                                 |    |
|---------------------------------|----|
| Introducción a PowerShell.....  | 1  |
| Fundamentos de PowerShell ..... | 7  |
| Control del Shell .....         | 10 |
| Conclusiones.....               | 10 |
| Bibliografía. ....              | 11 |

# Introducción a PowerShell

## Identificar las características de PowerShell



**PowerShell:** Es una solución de automatización de tareas multiplataforma formada por un shell de línea de comandos, un lenguaje de scripting y un marco de administración de configuración. Este término se refiere a la **interfaz de usuario**

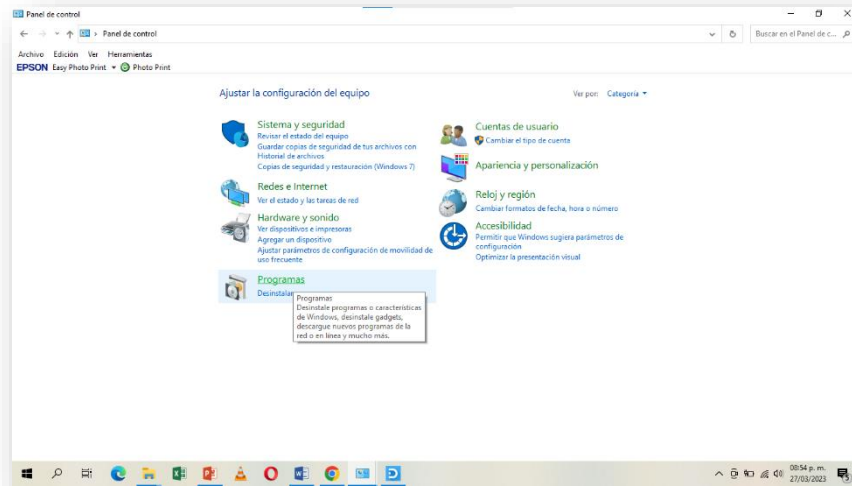
**visible** a través de la cual se puede interactuar con las funciones internas del sistema en un ordenador. PowerShell proporciona básicamente el mismo rango de comandos que cmd.exe. Si se conocen los comandos estándar más comunes se puede utilizar de la misma manera que la línea de comandos. Sin embargo, también permite el acceso a elementos internos más profundos de Windows a los que antes no se podía acceder con cmd.exe, por ejemplo, el Registry y la **Windows Management Instrumentation (WMI)**.

Las principales características de PowerShell son:

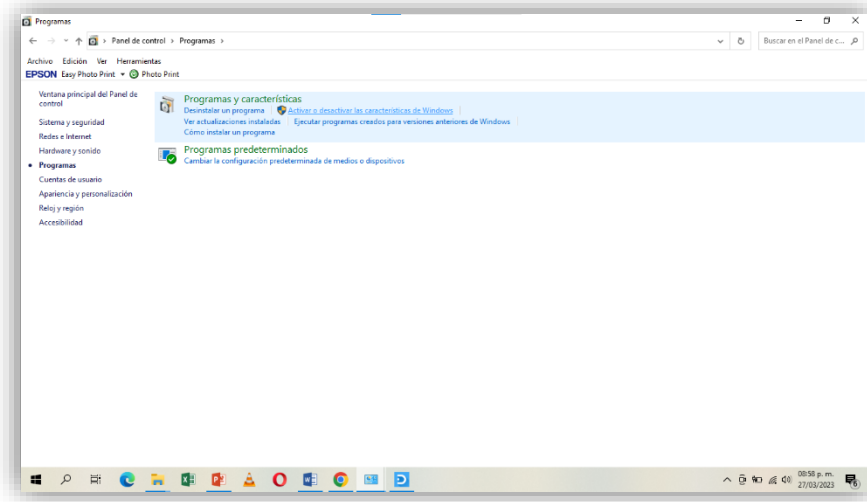
- Es un lenguaje de scripting.
- Diseñado para administradores de sistemas.
- Entorno interactivo y shell básica.
- Controlar y automatizar el sistema operativo.
- Permite controlar también aplicaciones de Windows.
- Incorpora conceptos y ventajas de distintos entornos.
- Realizar tareas de administración relacionadas con el registro, procesos, servicios, eventos, etc.
- En cuestión de seguridad tiene sistemas que controlan la ejecución de scripts y así se evita la ejecución de scripts no deseados.
- Las órdenes incluidas en PowerShell son muchas y reciben el nombre de cmd-lets.

# La instalación de PowerShell

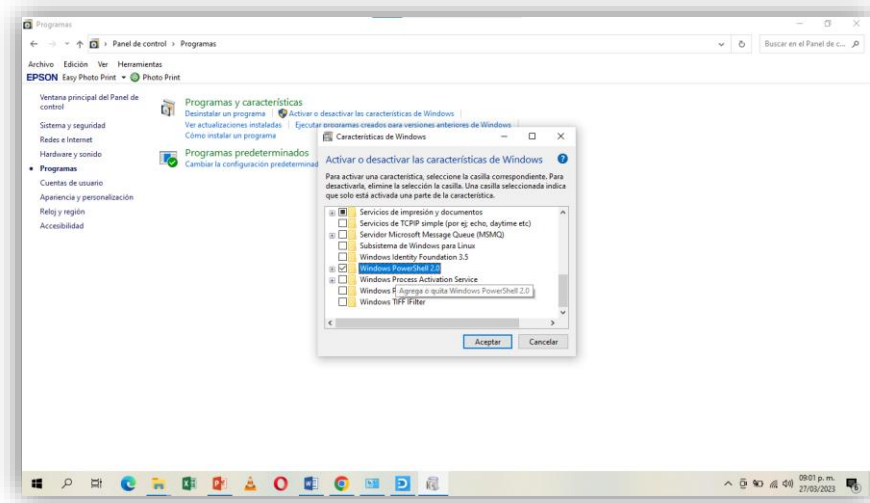
Lo primero que debemos hacer es ingresar al **Panel de control** y dirigirnos a **Programas**.



Ahora nos dirigimos a Activar o desactivar características.

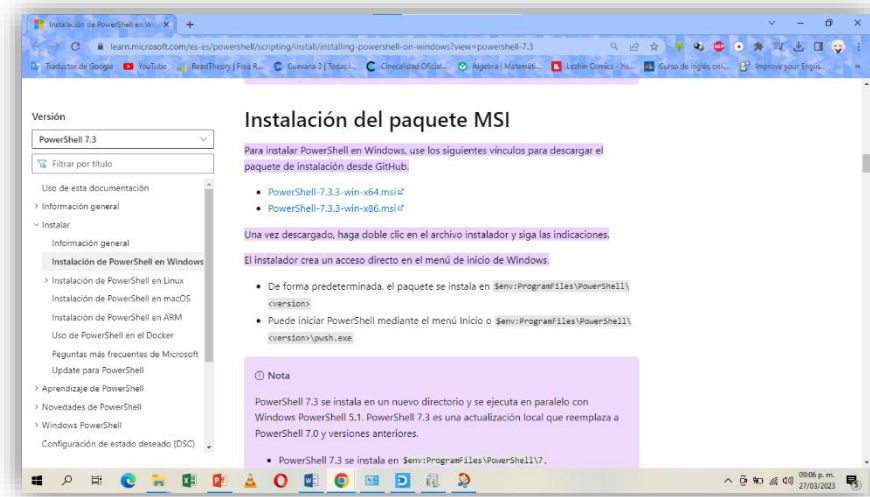


Buscamos PowerShell en caso de no tenerlo habilitado lo habilitamos.



Otra forma en la que podemos descargarlo es desde la página oficial de Microsoft y continuando con los pasos de instalación.

<https://learn.microsoft.com/es-es/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.3>



## Comandos básicos de PowerShell y su funcionalidad

El primer comando sirve para saber el número de comando disponible en PowerShell:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Tania> Get-Command -CommandType Cmdlet | Measure-Object

Count      : 553
Average    :
Sum        :
Maximum    :
Minimum    :
Property   :

PS C:\Users\Tania>
```

Otro comando simple es el de **dir** el cual muestra los directorios.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Tania> dir

Directorio: C:\Users\Tania

Mode                LastWriteTime         Length Name
----                -
d-----          29/09/2022   06:50 p. m.      .idlerc
d-----          27/03/2023   07:13 p. m.      .VirtualBox
d-----          01/08/2022   19:30 p. m.      .vscode
d-r-----        31/07/2022   02:22 p. m.      3D Objects
d-r-----        21/03/2023   10:07 p. m.      BrawlhallaReplays
d-r-----        08/03/2023   10:58 p. m.      Cisco Packet Tracer 8.2.0
d-r-----        31/07/2022   02:22 p. m.      Contacts
d-r-----        27/03/2023   04:06 p. m.      Desktop
d-r-----        25/03/2023   08:20 p. m.      Documents
d-r-----        27/03/2023   08:24 p. m.      Downloads
d-r-----        31/07/2022   02:22 p. m.      Favorites
d-r-----        31/07/2022   02:22 p. m.      Links
d-r-----        31/07/2022   03:50 p. m.      Music
d-r-----        31/07/2022   02:25 p. m.      OneDrive
d-r-----        21/03/2023   10:08 p. m.      Pictures
d-r-----        31/07/2022   02:22 p. m.      Saved Games
d-r-----        31/07/2022   02:23 p. m.      Searches
d-r-----        06/02/2023   09:17 p. m.      SecurityScans
d-r-----        02/03/2023   09:06 p. m.      Videos
d-r-----        27/03/2023   06:17 p. m.      VirtualBox VMs
d-r-----        16/03/2023   12:11 a. m.      wekafiles
-a-r-----        08/03/2023   10:55 p. m.      176 .packettracer

PS C:\Users\Tania>
```

Con este comando aparece lo mismo que el anterior, nos aparecen los directorios.

```
PS C:\Users\Tania> Get-ChildItem

Directorio: C:\Users\Tania
```

| Mode    | LastWriteTime          | Length | Name                      |
|---------|------------------------|--------|---------------------------|
| d-----  | 29/09/2022 06:50 p. m. |        | .idlerc                   |
| d-----  | 27/03/2023 07:13 p. m. |        | .VirtualBox               |
| d-----  | 01/08/2022 10:30 p. m. |        | .vscode                   |
| d-r---- | 31/07/2022 02:22 p. m. |        | 3D Objects                |
| d-----  | 21/03/2023 10:07 p. m. |        | BrawlhallaReplays         |
| d-----  | 08/03/2023 10:58 p. m. |        | Cisco Packet Tracer 8.2.0 |
| d-r---- | 31/07/2022 02:22 p. m. |        | Contacts                  |
| d-r---- | 27/03/2023 04:06 p. m. |        | Desktop                   |
| d-r---- | 25/03/2023 08:20 p. m. |        | Documents                 |
| d-r---- | 27/03/2023 08:24 p. m. |        | Downloads                 |
| d-r---- | 31/07/2022 02:22 p. m. |        | Favorites                 |
| d-r---- | 31/07/2022 02:22 p. m. |        | Links                     |
| d-r---- | 31/07/2022 03:50 p. m. |        | Music                     |
| d-r---- | 31/07/2022 02:25 p. m. |        | OneDrive                  |
| d-r---- | 21/03/2023 10:08 p. m. |        | Pictures                  |
| d-r---- | 31/07/2022 02:22 p. m. |        | Saved Games               |
| d-r---- | 31/07/2022 02:23 p. m. |        | Searches                  |
| d-----  | 06/02/2023 09:17 p. m. |        | SecurityScans             |
| d-r---- | 02/03/2023 09:06 p. m. |        | Videos                    |
| d-----  | 27/03/2023 06:17 p. m. |        | VirtualBox VMs            |
| d-----  | 16/03/2023 12:11 a. m. |        | wekafiles                 |
| -a----  | 08/03/2023 10:55 p. m. | 176    | .packettracer             |

El comando **Get-help** nos ayuda a obtener información acerca de un comando que deseemos conocer su función.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Tania> Get-Help Get-ChildItem

NOMBRE
    Get-ChildItem

SINTAXIS
    Get-ChildItem [[-Path] <string[]>] [[-Filter] <string>] [<CommonParameters>]

    Get-ChildItem [[-Filter] <string>] [<CommonParameters>]

ALIAS
    gci
    ls
    dir
```

El comando que se muestra a continuación es parecido al anterior solo que aquí se muestran ejemplos.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Tania> Get-Help Get-ChildItem -Examples

NOMBRE
    Get-ChildItem

ALIAS
    gci
    ls
    dir

NOTAS
    Get-Help no encuentra los archivos de Ayuda para este cmdlet en el equipo. Mostrará solo una parte de la Ayuda.
    -- Para descargar e instalar los archivos de Ayuda para el módulo que incluye este cmdlet, use Update-Help.
    -- Para ver en línea el tema de Ayuda de este cmdlet, escriba "Get-Help Get-ChildItem -Online" o
       vaya a https://go.microsoft.com/fwlink/?LinkID=113308.

PS C:\Users\Tania>
```

Este comando sirve para conocer la información detallada de cada argumento del comando.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Tania> Get-Help Get-ChildItem -Full

NOMBRE
    Get-ChildItem

SINTAXIS
    Get-ChildItem [[-Path] <string[]>] [[-Filter] <string>] [<CommonParameters>]
    Get-ChildItem [[-Filter] <string>] [<CommonParameters>]

PARÁMETROS
    -Attributes <FlagsExpression[FileAttributes]>

        {Requerido?           false
         {Posición?           Con nombre
         {Aceptar canalización? false
         Nombre de conjunto de parámetros (Todos)
         Alias                Ninguno
         {Dinámico?          true

    -Depth <uint32>

        {Requerido?           false
         {Posición?           Con nombre
         {Aceptar canalización? false
         Nombre de conjunto de parámetros (Todos)
         Alias                Ninguno
         {Dinámico?          false

    -Directory

        {Requerido?           false
         {Posición?           Con nombre
         {Aceptar canalización? false
         Nombre de conjunto de parámetros (Todos)
         Alias                ad, d
         {Dinámico?          true

    -Exclude <string[]>

        {Requerido?           false
         {Posición?           Con nombre
         {Aceptar canalización? false
```



# Fundamentos de PowerShell

## Sintaxis de PowerShell

Las interfaces como Cmd.exe no proporcionan al usuario una manera de ampliar directamente el conjunto de comandos integrados. Puede crear herramientas externas de línea de comandos que se ejecuten en Cmd.exe, pero estas herramientas externas carecen de servicios, como la integración de la Ayuda, y Cmd.exe no reconoce automáticamente que son comandos válidos.

Los comandos binarios nativos de Windows PowerShell, denominados cmdlets, se pueden ampliar con cmdlets que cree y que agregue a Windows PowerShell mediante complementos. Los complementos de Windows PowerShell se compilan, al igual que las herramientas binarias de cualquier otra interfaz. Puede utilizarlos para agregar proveedores de Windows PowerShell al shell, así como nuevos cmdlets.

Debido a la naturaleza especial de los comandos internos de Windows PowerShell, los llamaremos cmdlets.

Windows PowerShell incluye palabras clave y funciones de sintaxis muy parecidas a las que se usan en el lenguaje de programación C#, ya que también se basa en .NET Framework. Aprender a utilizar Windows PowerShell facilita mucho el aprendizaje de C#, si está interesado en este lenguaje.

Si no es un programador de C#, esta similitud no es importante. No obstante, si ya está familiarizado con C#, las similitudes pueden facilitar enormemente el aprendizaje de Windows PowerShell.

## Características de los argumentos y parámetros de PowerShell

los nombres de los parámetros de los comandos utilizados en las interfaces tradicionales de línea de comandos no suelen ser coherentes. Algunos parámetros no tienen nombre. Si tienen nombre, suele ser una palabra abreviada o de un solo carácter que se pueden escribir rápidamente, pero que los usuarios nuevos no entienden fácilmente.

A diferencia de la mayoría de las interfaces tradicionales de línea de comandos, Windows PowerShell procesa los parámetros directamente y usa este acceso directo a los parámetros, junto con las directrices del programador, para normalizar los nombres de parámetros. Aunque esto no garantiza que todos los cmdlets se ajusten siempre a los estándares, sí lo fomenta.

- **Parámetro Help (?)**: Cuando se especifica el parámetro -? en cualquier cmdlet, no se ejecuta el cmdlet, sino que se muestra la Ayuda correspondiente.

- **Parámetros comunes:** Windows PowerShell incluye varios parámetros conocidos como los parámetros comunes. Dado que se controlan mediante el motor de Windows PowerShell, estos parámetros se comportan de la misma manera siempre que un cmdlet los implementa. Los parámetros comunes son **WhatIf**, **Confirm**, **Verbose**, **Debug**, **Warn**, **ErrorAction**, **ErrorVariable**, **OutVariable** y **OutBuffer**.
- **Parámetros recomendados:** Los cmdlets principales de Windows PowerShell utilizan nombres estándar para parámetros similares. Aunque el uso de nombres de parámetros no es obligatorio, existen unas directrices de uso explícitas a fin de fomentar la normalización.

Por ejemplo, estas directrices recomiendan llamar **ComputerName** a un parámetro que haga referencia a un equipo por su nombre, en lugar de **Server**, **Host**, **System**, **Node** u otras palabras alternativas comunes. Algunos nombres de parámetros recomendados importantes son **Force**, **Exclude**, **Include**, **PassThru**, **Path** y **CaseSensitive**.

## Proceso de creación de objetos personalizados

Puede crear objetos en PowerShell y usar los objetos que cree en comandos y scripts:

- ❖ **New-Object:** crea una instancia de un objeto de .NET Framework o un objeto COM.
- ❖ **Import-Csv/ConvertFrom-CSV:** crea objetos personalizados (PSCustomObject) a partir de los elementos definidos como valores separados por caracteres.
- ❖ **ConvertFrom-Json:** crea objetos personalizados definidos en notación de objetos JavaScript (JSON).
- ❖ **ConvertFrom-StringData:** crea objetos personalizados definidos como pares clave-valor.
- ❖ **Tipo de complemento:** permite definir una clase en la sesión de PowerShell que puede crear instancias con **New-Object**.
- ❖ **New-Module:** el parámetro **AsCustomObject** crea un objeto personalizado que se define mediante el bloque de script.
- ❖ **Add-Member:** agrega propiedades a objetos existentes. Puede usar **Add-Member** para crear un objeto personalizado a partir de un tipo simple, como `[System.Int32]`.

- ❖ **Select-Object:** selecciona las propiedades de un objeto. Puede usar Select-Object para crear propiedades personalizadas y calculadas en un objeto ya creado.



# Control del Shell

## Proceso de personalización del entorno Shell

Un perfil de PowerShell es un script que se ejecuta cuando se inicia PowerShell. Puede usar el perfil para personalizar el entorno. Puede:

- Agregar alias, funciones y variables
- Cargar módulos
- Crear unidades de PowerShell
- Ejecutar comandos arbitrarios
- Cambiar la configuración de preferencias

Aplicar esta configuración en el perfil garantiza que esté disponible cada vez que inicie PowerShell en el sistema.

## Estructura de los comandos

Los nombres de todos los cmdlets siguen un patrón común formado por un verbo, un guion y un nombre en singular.

- PowerShell no distingue entre mayúsculas y minúsculas.
- Dispone de autocompletado si usamos ISE.
- Argumentos y operadores.
- Opcionales y obligatorios.

## Conclusiones

PowerShell es una herramienta muy útil ya que nos ayuda a automatizar la administración de sistemas, también a compilar, probar e implementar soluciones con una variedad de comandos y scripts seguros con base en certificados y un funcionamiento muy estricto para la resolución de problemas y de igual manera nos proporciona experiencia en el uso de línea de comandos integrada en el sistema operativo

## **Bibliografía.**

Hacking, E. (2022). Como Descargar e instalar nueva versión de Windows PowerShell 7[Video]. Obtenido de <https://www.youtube.com/watch?v=QJreo9Cifkl>

Ionos. (27 de febrero de 2019). Digital Guide. Obtenido de <https://www.ionos.mx/digitalguide/servidores/know-how/windows-powershell/>

Microsoft. (19 de marzo de 2023). Instalación de PowerShell en Windows. Obtenido de <https://learn.microsoft.com/es-es/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.3>

Microsoft. (10 de enero de 2023). Personalización del entorno de shell. Obtenido de <https://learn.microsoft.com/es-es/powershell/scripting/learn/shell/creating-profiles?view=powershell-7.3>

Microsoft. (13 de diciembre de 2022). about\_Object\_Creation. Obtenido de [https://learn.microsoft.com/es-es/powershell/module/microsoft.powershell.core/about/about\\_object\\_creation?view=powershell-7.3](https://learn.microsoft.com/es-es/powershell/module/microsoft.powershell.core/about/about_object_creation?view=powershell-7.3)