

# Semana 08:

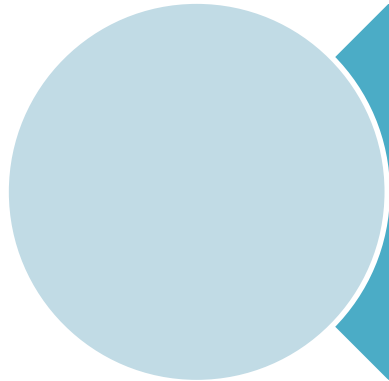
## Introducción a PowerShell, Scripts

Sistemas Operativos

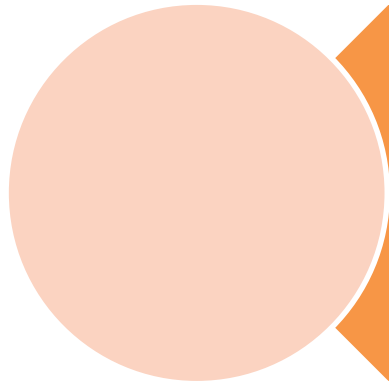
# Introducción

- Windows PowerShell es un shell de línea de comandos y un entorno de scripting que ofrece la eficacia de .NET Framework a los usuarios de la línea de comandos y a los creadores de scripts.
- Windows PowerShell presenta diversos conceptos nuevos y eficaces que permiten ampliar los conocimientos adquiridos y los scripts creados en los entornos del símbolo del sistema de Windows.

# Capacidades de la Sesión



Manejar el entorno  
Windows PowerShell.



Implementar scripts  
en PowerShell.

# 1. Introducción a Windows PowerShell

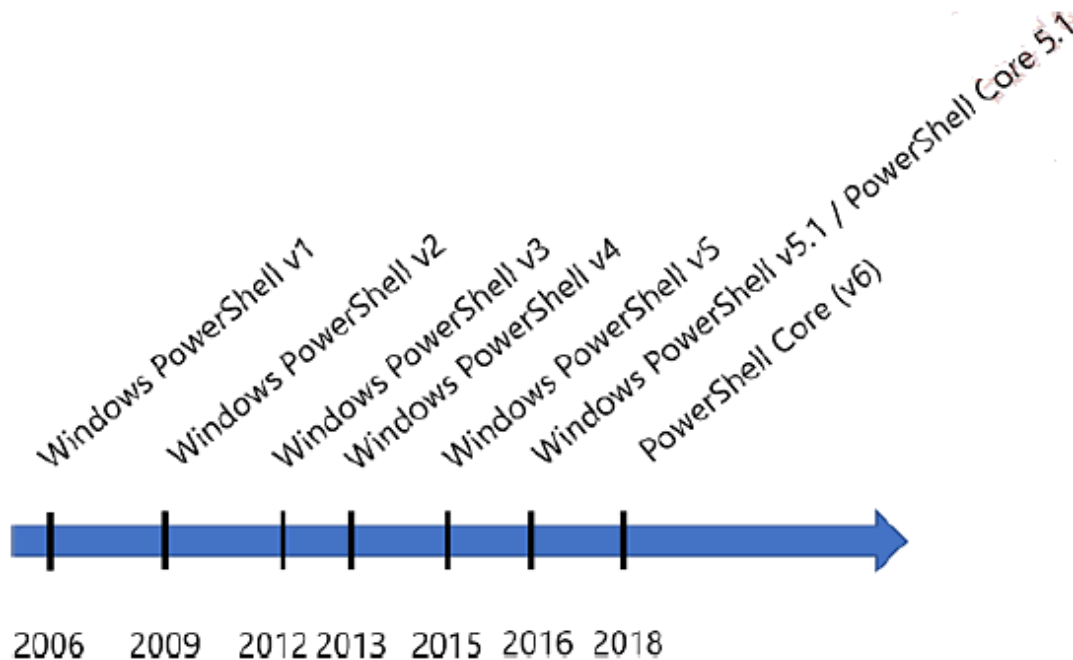
- Windows PowerShell es un nuevo shell de línea de comandos y lenguaje de scripting basado en tareas diseñado especialmente para la administración del sistema.
- Basado en .NET Framework
- Windows PowerShell ayuda a los profesionales de TI y usuarios avanzados a controlar y automatizar la administración del sistema operativo Windows y las aplicaciones que se ejecutan en Windows.

# 1. Introducción a Windows PowerShell

- Los administradores de sistemas utilizan scripts para automatizar las tareas pesadas.
- El tiempo invertido en desarrollar un script solo es rentable si pasamos menos tiempo haciendo el script que realizando la tarea manualmente.
- Ejm: Creación de 100 cuentas de usuario en AD.

# 1.1. Conceptos Básicos de Windows PowerShell

- PowerShell es a la vez un intérprete de comandos y un potente lenguaje de scripts.
- Basado en Framework .NET.
- Windows PowerShell utiliza la sintaxis del lenguaje C#.
- PSCore nace en el 2016 y se basa en .NET Core (open source).




## 1.2. Características de Windows PowerShell


- 129 cmdlets estándar para realizar tareas comunes de administración del sistema.
- Un lenguaje de scripting basado en tareas y soporte para scripts y herramientas de línea de comandos existentes.
- Navegación simplificada basada en comandos del sistema operativo, que permite a los usuarios navegar por el Registro y otros almacenes de datos.
- Grandes capacidades de manipulación de objetos.

## 1.2. Características de Windows PowerShell (cont.)

- Para ver la cantidad de módulos (cmdlet,function,alias) instalados en nuestro powershell podemos apoyarnos con el comando

“(get- Administrador: Windows PowerShell  
PS C:\> (get-command).count  
2002

- Para ver la versión de nuestro powershell podemos usar los comandos “get-host” o “\$PSversiontable”.

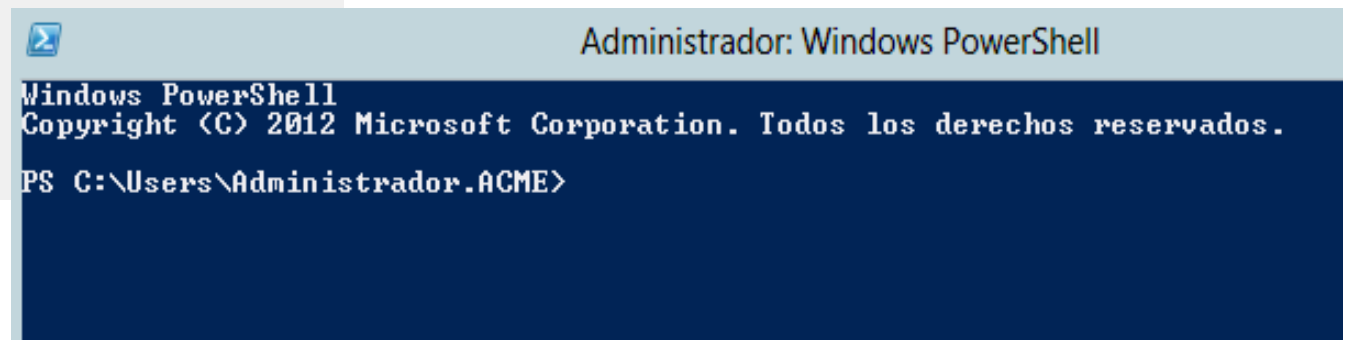
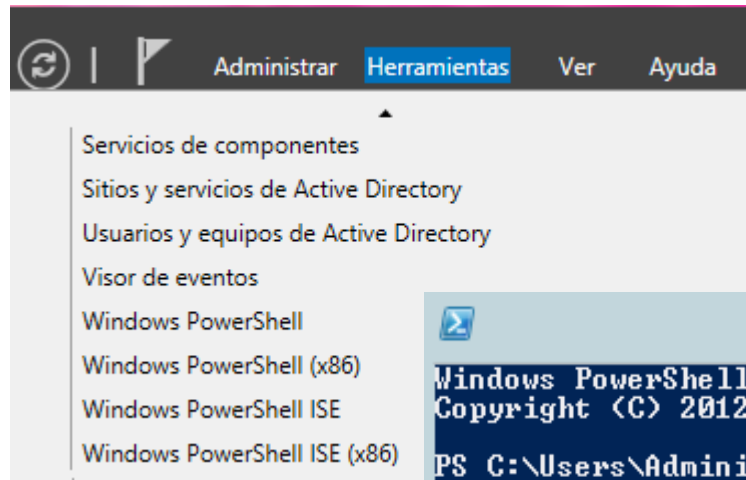
 Administrador: Windows PowerShell  
PS C:\> get-host

Name	: ConsoleHost
Version	: 5.1.14393.0
InstanceId	: 28c465a6-eea1-4978-b0a0-11df52ef110d
UI	: System.Management.Automation.Internal.Host.InternalHost
CurrentCulture	: es-PE
CurrentUICulture	: es-ES
PrivateData	: Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
DebuggerEnabled	: True
IsRunspacePushed	: False
Runspace	: System.Management.Automation.Runspace.LocalRunspace



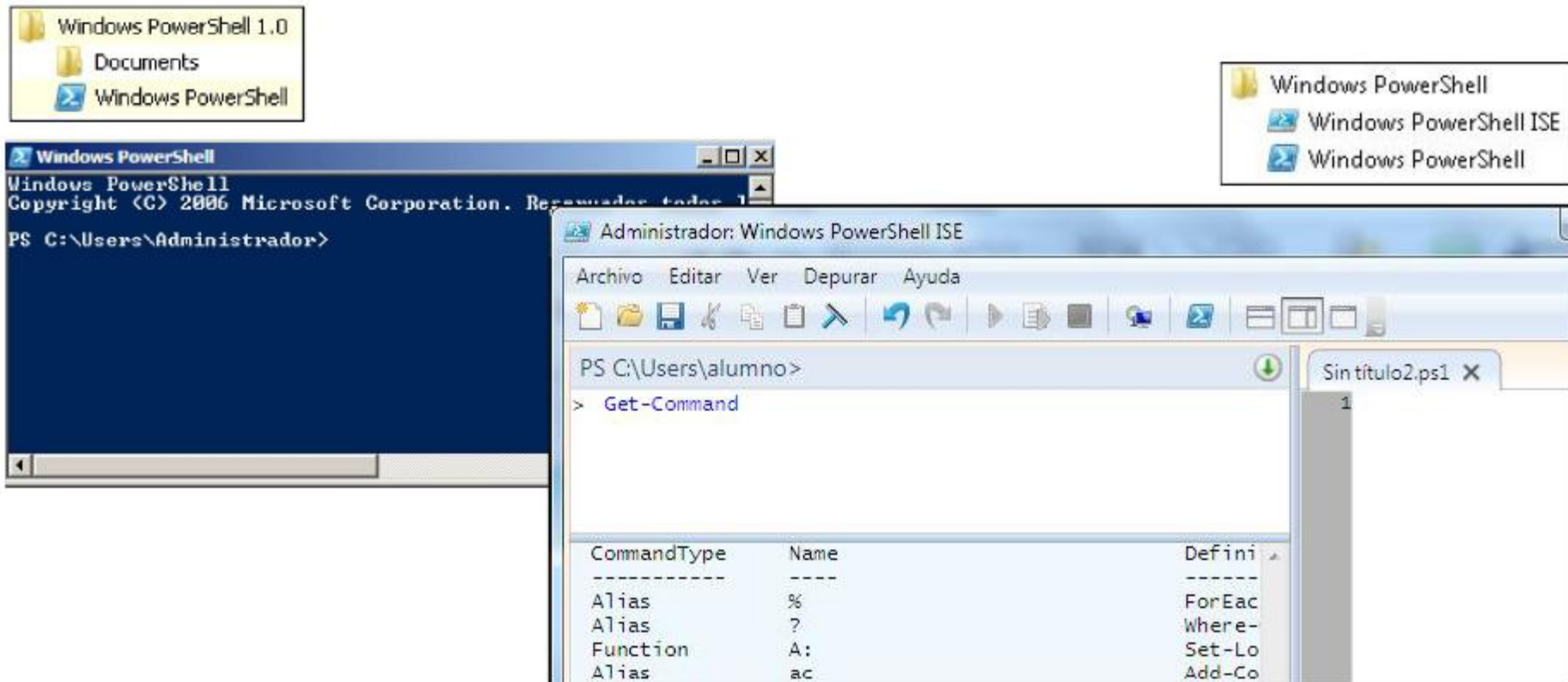
## 1.3. Ejecutar Windows PowerShell

- Desde la barra de tareas, botón Windows PowerShell.
- Desde el Server Manager -> Herramientas -> Windows PowerShell.



## 1.3. Ejecutar Windows PowerShell

- Existe dos modos:
  - CLI (Command Line Interface)
  - ISE (Integrated Scripting Environment)



## 2. Instrucciones Disponibles en Windows PowerShell

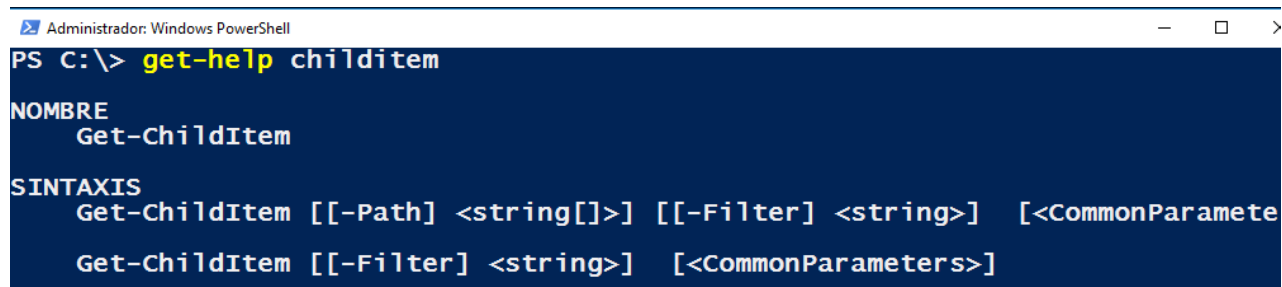
- Tipos de instrucciones:
  - cmdlet: comandos individuales, usa parámetros: get, set, format, stop, etc.
  - alias: nombre sencillo (alternativo) para cmdlet o function.
  - function: es un script (similar a un cmdlet).
- Se puede visualizar con:
  - Get-command
  - Get-Command -CommandType cmdlet
  - Get-Command -CommandType alias
  - Get-Command -CommandType function

# Instrucciones Disponibles

Cmdlet	Alias	Description
Get-Location	cd	Display the current directory
Set-Location	cd, chdir	Change the current directory
Clear-Host	cls	Clear the screen
Copy-Item	copy	Copy one or several files
Get-Help	help	Help on commands
Remove-Item	del, rmdir, rd	Delete a file / a directory
Rename-Item	ren, rename	Rename a file / a directory
Move-Item	move	Move a file / a directory to a new location
Get-ChildItem	dir	List all files / directories in the directory
Write-Output	echo	Print strings, variables etc. to standard output
Get-Content	type	Get the content of a file
Get-Process	tlist, tasklist	List all currently running processes
Stop-Process	kill, taskkill	Stop a running process

## 2.1. Comandos de Ayuda

- Obtener información de ayuda detallada
  - `get-help get-childitem`
  - `get-childitem -?`
  - `help get-childitem`



```
Administrador: Windows PowerShell
PS C:\> get-help childitem

NOMBRE
    Get-ChildItem

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

## 2.2. Alias

- Obtener el cmdlet de un alias.
  - Get-Alias cls
- Crear nuevos alias (ejemplos).
  - Set-Alias -Name agregar-rol -Value add-windowsfeature
  - Set-Alias -Name ver-procesos -Value get-process

```
Administrador: Windows PowerShell  
PS C:\> set-alias -name ver-procesos -value get-process
```

## 2.3. Comodines

- PowerShell permite la utilización de comodines
  - `Get-Command -name Get-Pro*`
  - `Get-Command -name [A-D]*`
  - `Get-Command -name ??[E-G][A]*`

## 2.4. Mostrando los resultados

- Salidas de los resultados:
  - En la pantalla
  - En una variable
  - En un archivo
- Ejemplo:
  - Get-process
  - \$a=Get-process
  - \$a
  - Get-process > archivo.txt
  - type archivo.txt



## 2.5. Redirección y Pipelines

- Uso de pipelines:
  - >
  - >>
  - |
- Ejemplo:
  - Get-process > archivo.txt
  - Get-process >> archivo.txt
  - Get-process | format-list

### 3. Manipulando Elementos

- Un elemento o “Item” puede representar, archivos, carpetas, claves del registro, certificados, variables de entorno.
- Podemos enumerar todos los elementos de un contenedor o “Drive” con Get-Childitem.
- Cmdlets para manipular elementos
  - New-Item
  - Remove-Item
  - Copy-Item
  - Rename-Item
  - Move-Item
  - Set-Item
  - Get-Item



## 3.1. Trabajar con archivos, carpetas y claves del Registro

- Windows PowerShell utiliza el sustantivo Item para referirse a elementos que se encuentran en una unidad de Windows PowerShell.
- Cuando utilice el proveedor FileSystem de Windows PowerShell, un Item puede ser un archivo, una carpeta o la unidad de Windows PowerShell.

# 3.1. Trabajar con archivos, carpetas y claves del Registro

Common File System Command
Dir
Move
Ren
Del, Rmdir
Copy
Mkdir
Cd

Windows PowerShell cmdlet
Get-ChildItem
Move-Item
Rename-Item
Remove-Item
Copy-Item
New-Item
Set-Location
Get-Location
Get-ItemProperty
Set-Item
Set-ItemProperty

## 3.2. Enumerar archivos, carpetas y claves del Registro

- Ejemplos:
  - `Get-ChildItem -Path C:\Windows`
- Filtrar elementos por nombre (-Name)
  - `Get-ChildItem -Path C:\WINDOWS -Name notepad.exe`
- Forzar la presentación de los elementos ocultos (-Force)
  - `Get-ChildItem -Path C:\Windows -Force`
- Usar caracteres comodín
  - `Get-ChildItem -Path C:\Windows\?????.log`
  - `Get-ChildItem -Path C:\Windows\x*`
  - `Get-ChildItem -Path C:\Windows\[x:z]*`

## 3.3. Obtener la ubicación actual (Get-Location)

- Ejemplo:
  - Get-Location
- Establecer la ubicación actual (Set-Location)
  - Set-Location -Path C:\Windows

```
Administrador: Windows PowerShell
PS C:\> get-location

Path
----
C:\

PS C:\> Set-Location -Path C:\Windows\System32\drivers\etc\
PS C:\Windows\System32\drivers\etc> get-location

Path
----
C:\Windows\System32\drivers\etc
```

## 4. SCRIPTS con PowerShell

- Ficheros de texto con extensión \*.ps1
- Se ejecutan:
  - Directamente en la consola de PowerShell
  - Mediante el siguiente comando:
    - PowerShell.exe c:\scripts\Script.ps1
- Con el modificador –noexit no se cerrará la consola después de la ejecución

## 4.1. Variables

- Las variables empiezan por \$ y no tienen tipos específicos por defecto, aunque pueden ser declaradas explícitamente
  - `$a=10`
  - `$b= "Esto es una cadena"`
  - `$c= 1,2,3`
  - `$d= 1..10`
  - `$e="Esto es", "un array", "de cadenas"`
  - `$f= (1,2),(3,4),(5,6)`
  - `[int]$entero=1`
  - `[string]$cadena="Esto es otra cadena"`
- Las variables creadas en una sesión de PowerShell solo perduran durante esa sesión, y las creadas en un script solo durante la ejecución del script.



## 4.2. Solicitar ingreso de datos

- Los cmdlets Read-Host y Write-Host son útiles a la hora de pedir y devolver valores durante la ejecución de un Script
  - \$nombre = Read-Host “Introduce tu Nombre”
- Las variables también son objetos, y por tanto pueden almacenarlos.

## 4.3. Bucles y Condicionales

- Soporta los siguientes bucles:
  - While, Do..While, Until: Ejecuta código mientras la condición sea cierta.
  - For: Ejecuta código un número especificado de veces.
  - ForEach: Ejecuta código para cada elemento de una colección.
- Expresiones condicionales
  - If .. Else .. Elseif
  - Switch



## 4.4 Políticas de Ejecución

- Determinan:
  - Si se pueden o no ejecutar scripts y bajo que condiciones de seguridad.
  - Si se pueden cargar ficheros de configuración personalizados.
- Cuatro estados:
  - Restricted: (defecto): No se permite la ejecución de scripts.
  - AllSigned: Los scripts deben estar firmados digitalmente por un emisor de confianza para poder ejecutarse.
  - RemoteSigned: Los scripts que se hayan descargado, que vengan por correo electrónico, han de estar firmados digitalmente por un emisor de confianza para poder ejecutarse. Los locales se ejecutarán todos.
  - Unrestricted: Todos los scripts podrán ejecutarse.

# Preguntas de Repaso:

¿Cuál es comando para crear un directorio en la raíz de la unidad C?



¿Cuál es el comando PowerShell para los alias de Dir, Cd, Copy, Del, Move y



# FIN DE LA UNIDAD

# Bibliografía

- VanJones, M & Deman , T & Elmale, F. & Desfarges, G (2018) Windows Server 2016 Administración avanzada. Barcelona: Ediciones ENI
- Carretero Pérez, Jesús (2001). Sistemas operativos. Una visión aplicada. Madrid: Mc Graw-Hill (005.43/C28)
- Charre Ojeda, Francisco (2008). Windows Server 2008. Madrid: Anaya Multimedia (005.43WI/C525)
- Raya Gonzalez, Laura. (2005).Sistemasoperativos en entorno monousuarios y multiusuarios . México D.F.: Alfaomega (005.43/R28R)

