

# PRIMERA ECTS DE FC

## PowerShell y el uso de Git en Github



Nombre: Jose Luís Obiang Ela Nanguan

Profesor: Francisco Fernandez de Vega

Fecha 27/04/2022

Desarrollado por José Luis Obiang Ela Nanguan Version Actual 27.04.2022

Puedes obtener toda la documentación aquí [JLOEN1999](#)

## Índice de contenido

1. **Presentación**
  - 1.1. **Objetivos**
2. **PowerShell**
  - 2.1. **Introducción**
  - 2.2. **La línea de comandos**
  - 2.3. **¿Qué es PowerShell?**
  - 2.4. **Abrir la consola de PowerShell**
  - 2.5. **La ayuda en PowerShell**
  - 2.6. **Gestión de archivos y carpetas**
  - 2.7. **Tuberías y redireccionamiento**
  - 2.8. **Iniciación a los scripts**
  - 2.9. **Fundamentos de scripts-I: Variables**
  - 2.10. **Fundamentos de scripts-II: Estructuras de control y funciones**
3. **Github**

### 1. Presentación

Este curso está planteado en lo más práctico posible. Las Prácticas que vamos a realizar son las siguientes:

*En el primer momento empezaremos a trabajar con la PowerShell y la PowerShell ISE*

*Vamos a buscar Información*

*Vamos a utilizar los comandos básicos relacionados con la gestión de archivos y carpetas*

***Vamos a enlazar la salida de un comando con la entrada de otro y redireccionar la salida***

***Y Vamos a realizar pequeños scripts y hablando de Scripts, vamos a hacer un script en el que combinaremos estructuras condicionales, repetitivas y redireccionamiento.***

\_\_También haremos uso de la herramienta Git que es un sistema de control de versiones desarrollado por Linux Torvalds en el año 2005 y que se ha hecho muy popular gracias a servicios como Github.\_\_Si eres programador, desarrollador web, administrador de sistemas, diseñador, ... es muy probable que en algún momento de tu trabajo te encuentres con un proyecto en el que tengas que colaborar con otras personas usando Git. Puede que trabajes solo pero que te interese tener un seguimiento y control de tu trabajo. Es estos dos casos y en muchos más un conocimiento más o menos profundo de Git te permitirá ser mucho más productivo en tu trabajo y, sobre todo, evitar muchos problemas con los que se encuentra a menudo la gente que no trabaja con un sistema de control de versiones.

Si tu ámbito de trabajo es técnico y aún no usas Git, cuando lleves unos meses usándolo te preguntarás cómo es posible que no lo hubieras empezado a usar antes.

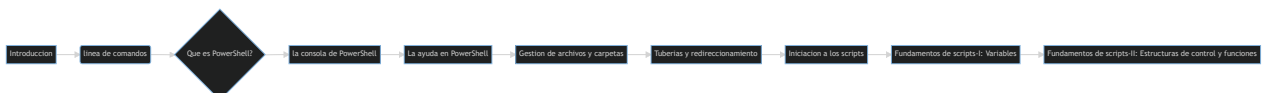
## 1.1. Objetivos

***¿Qué vamos a conseguir al finalizar este curso?***

Pues vamos a:

<b>A manejar tanto la PowerShell como la PowerShell ISE con soltura.</b>	<b>Buscar información en la ayuda de PowerShell.</b>	<b>Conocer los comandos básicos.</b>	<b>Ser capaces de realizar scripts para automatizar determinadas tareas.</b>	<b>Conocer los comandos básicos de Git para hacer un buen seguimiento de nuestros proyectos en Github.</b>
--	--	--------------------------------------	--	--

## 2. PowerShell



### 2.1. Introducción

La interfaz de usuario es el medio que utilizamos para comunicarnos con el ordenador.

Interfaz gráfica: GUI(Proporciona un entorno visual)

Interfaz de línea de comandos:CLI(Command Line Interface, nos permite dar instrucciones por medio de una línea de texto)

### ¿Cuál de las dos debemos utilizar?

La respuesta es fácil, depende de lo que queramos hacer, si quieres navegar, trabajar con un procesador de texto, hoja de cálculo, retocar fotografía, etc, tu respuesta es la interfaz gráfica. Y si lo que quieres es automatizar tareas, crear usuarios de forma masiva, comprobar conectividad con servidores pues la respuesta es PowerShell(La línea de comandos)

## 2.2. La línea de comandos

Vamos a ver ahora cómo ha evolucionado la línea de comandos de Windows

### ***CMD o símbolo del sistema:***

Todavía convive con nosotros pero cada vez se utiliza menos, tal vez para hacer un ping o ipconfig

ping 8.8.8.8 (Para comprobar si nuestra PC tiene conexión a Internet)

```
C:\Users\JLOel>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=10ms TTL=118
Respuesta desde 8.8.8.8: bytes=32 tiempo=12ms TTL=118
Respuesta desde 8.8.8.8: bytes=32 tiempo=13ms TTL=118

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 3, recibidos = 3, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 10ms, Máximo = 13ms, Media = 11ms
```

ipconfig (Para ver los adaptadores de red de la PC)

---

```
C:\Users\JLOel>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet vEthernet (Wi-Fi):

    Sufijo DNS específico para la conexión. . :
    Vínculo: dirección IPv6 local. . . : fe80::98a:306a:99ce:84de%42
    Dirección IPv4. . . . . : 172.19.160.1
    Máscara de subred . . . . . : 255.255.240.0
    Puerta de enlace predeterminada . . . . . :
```

## PowerShell

En cambio la PowerShell se pensó como una herramienta de reemplazo del CMD y con el tiempo se ha convertido en una herramienta poderosa de gestión tanto para usuarios domésticos como administradores

### 2.3. ¿Qué es PowerShell?

La PowerShell es una nueva línea de comandos, es decir, es una herramienta multiplataforma utilizada principalmente por los administradores de Sistemas Windows para automatizar tareas y tener un mayor control del sistema. Esta herramienta está formada por una shell de comandos, un lenguaje de scripting y un marco de administración de configuración. Trabaja con objetos, acepta y devuelve objetos en vez de texto como ocurre con el CMD

**¿Dónde podemos encontrar PowerShell?** En Windows 10 la encontramos, Windows Server, Microsoft Azure, SQL Server, Sercivios de Office 365, se encuentra prácticamente en todos los productos de Microsoft

Versiones	
Versión	año
V1	2006
V2	2009
V3	2012
V4	2013
V5	2016
V5.1	2017
V Core 6.0	2018

No tenemos que confundir:

- Windows PowerShell ISE, es un entorno en el que podemos ejecutar comandos, escribir, probar y depurar script.
- Windows PowerShell es la consola de comandos.

**¿Qué requisitos se necesitan para aprender dicha herramienta?** Como se trata de un curso a nivel de iniciación en PowerShell, cualquier persona con conocimientos de informática a nivel de usuario podría hacerlo sin mayor problema, ahora bien hay una parte en la que se habla de variables y estructuras condicionales y entonces aquí si se requiere conocimientos mínimos de programación.

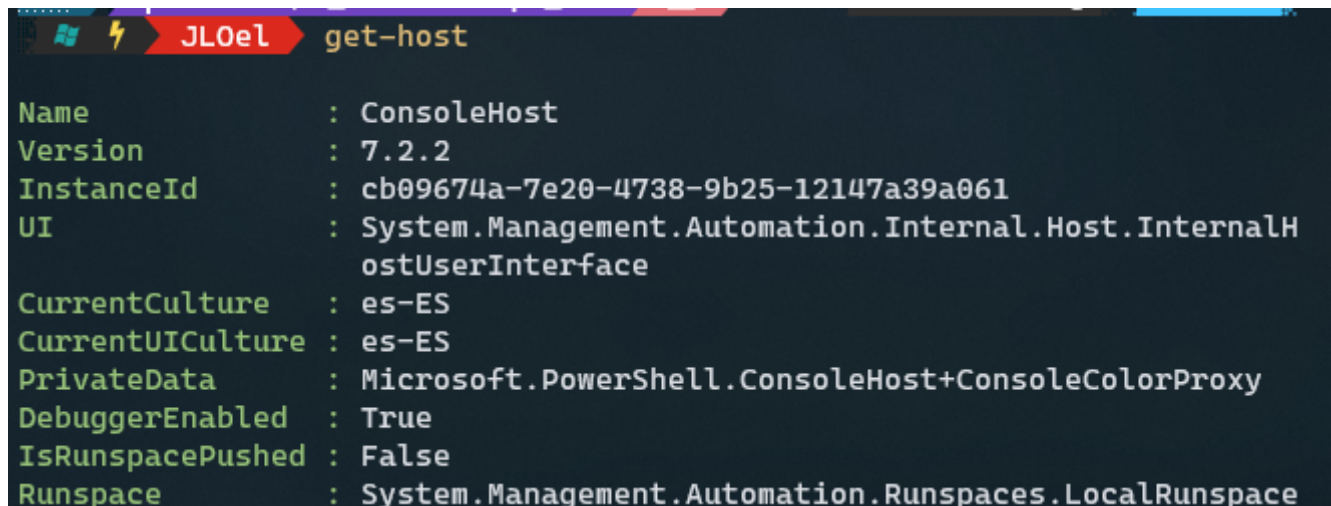
**¿Qué máquina necesitamos para trabajar en PowerShell?** Es suficiente con tener un Windows 10 instalado o bien un Windows Server

## 2.4. Abrir la consola de PowerShell

Hay varias maneras de abrir la consola de comandos en Windows:

- Dando click derecho sobre el símbolo de Windows y pinchamos donde aparece PowerShell
- Pulsando Windows+R y escribimos `PowerShell`

Podemos ver la versión que tiene nuestro PowerShell con el comando `get-host`



```
PS C:\> get-host

Name       : ConsoleHost
Version    : 7.2.2
InstanceId : cb09674a-7e20-4738-9b25-12147a39a061
UI         : System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture   : es-ES
CurrentUICulture : es-ES
PrivateData     : Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
DebuggerEnabled : True
IsRunspacePushed : False
Runspace        : System.Management.Automation.Runspaces.LocalRunspace
```

## 2.4. La ayuda en PowerShell

La ayuda en PowerShell es muy completa y trae muchos ejemplos, necesitamos acceso a internet para descargarla, eso puede parecer una limitación pero nos permite tener una ayuda actualizada. Para actualizar la ayuda de PowerShell necesitamos el comando `update-help` y eso para tener actualizada las últimas novedades, la ayuda de PowerShell nos da muchos ejemplos del uso de cada comando



```
PS C:\> update-help
Updating Help for module Microsoft.PowerShell.Management [Locating Help Content ...]
```

**Concepto de cmdlet:** En verdad en PowerShell se habla de cmdlet(commandlet). Es una combinación de verbo y nombre separados por un guión (-). **Verbo:** describe la acción que se va a realizar **nombre** es

el objeto sobre el que se va a realizar la acción. Para entender mejor ahora abrimos nuestra consola:

- Ejemplos de Verbos: `get` (Obtiene una información) `set` (Cambia una propiedad) `remove` (Elimina un objeto) `new` (Crea un objeto) pero estos verbos no hacen nada si no les aplicamos un nombre

```
PS master # +165 -8 | +7 7:14:37 PM 104ms
JLOel get set remove new pwsh
```

- Ejemplos de Nombres: `localuser` `localgroup` `netadapter` `partition`, estos nombres tampoco hacen algo si no les aplicamos un verbo

```
PS master # +165 -8 | +7 7:14:37 PM 104ms
JLOel localuser localgroup netadapter partition pwsh
```

Vamos a combinarlos a ver si ahora conseguimos algo. Ejemplos: `Get-localuser` (Muestra todos los usuarios locales del sistema).

```
PS master # +165 -8 | +7 7:14:37 PM 104ms
JLOel get-localuser pwsh

Name                Enabled Description
----                -
Administrador       True   Cuenta integrada para la administración del...
DefaultAccount      False  Cuenta de usuario administrada por el siste...
Invitado            False  Cuenta integrada para el acceso como invita...
JLOel               True   ...
WDAGUtilityAccount  False  Una cuenta de usuario que el sistema admini...
```

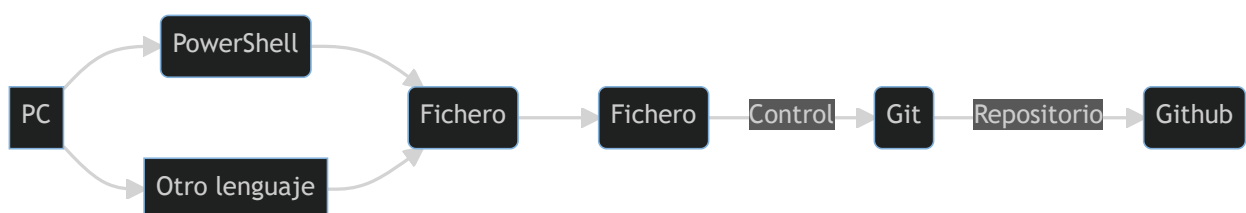
NOTA: Una cosa, PowerShell no es sensible a las mayúsculas ni minúsculas, es decir, puedes poner un comando todo mayúsculas o minúsculas o puedes acceder a un archivo de igual manera `get-date` (Para saber la fecha actual del sistema)

```
PS master # +165 -8 | +7 9:48:28 PM 300ms
JLOel get-date pwsh

miércoles, 27 de abril de 2022 21:51:28
```

`clear` (Para limpiar la pantalla)

### 3. Git en Github



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eaque esse harum modi omnis veritatis. Accusantium, architecto, aspernatur assumenda blanditiis commodi cum cumque cupiditate dignissimos dolore enim facere fugit harum incidunt maiores maxime minima modi natus nostrum nulla numquam, quasi ratione repellat sint totam velit voluptas voluptate voluptates! Ab asperiores at cupiditate dicta dolore dolorem dolores dolorum ea, eaque earum, error est hic, illum ipsa magni maxime minus nesciunt nostrum odit officia quae quam quas quidem quis quo reprehenderit tempora tempore. Animi corporis ea eligendi hic mollitia, nulla!

