

# Pipeline, variables, estructuras de control, scripts y funciones

DigitalHouse>



**Certified Tech  
Developer**

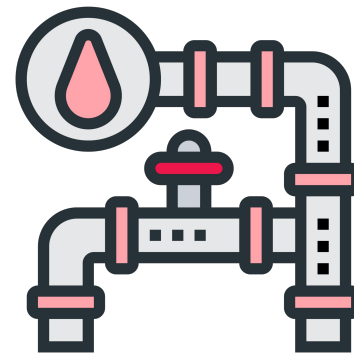
The Ultimate Degree

# ¿Qué es un pipeline?

Es una serie de comandos conectados mediante operadores de canalización ("|", ASCII 124). Cada operador del pipeline envía los resultados del comando anterior al siguiente comando.

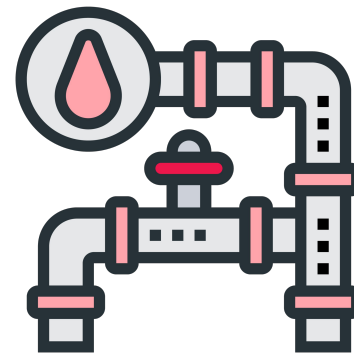
La salida del primer comando se puede enviar para procesar como entrada para el segundo comando y el resultado se puede enviar a otro comando. Lo anterior es una cadena de comandos o pipeline complejo que se compone de una serie de comandos simples.

Por ejemplo: Command-1 | Command-2 |  
Command-3



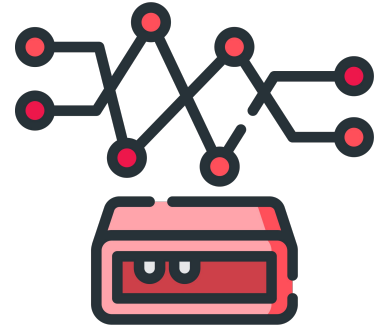
# ¿Qué es un pipeline?

En este ejemplo, los objetos que Command-1 emite se envían a Command-2 . Command-2 procesa los objetos y los envía a Command-3. Este último procesa los objetos y los envía a través del pipeline. Dado que no hay más comandos en la canalización, los resultados se muestran en la consola. En un pipeline, los comandos se procesan en orden de izquierda a derecha. El procesamiento se controla como una operación única y el resultado se muestra a medida que se genera.



# ¿Qué es una variable?

Una variable es una unidad de memoria en la que se almacenan los valores. En PowerShell, las variables se representan mediante cadenas de texto que comienzan por un signo de dólar (\$), como **\$a**, **\$process** o **\$my\_var**.



Los nombres de variable no distinguen mayúsculas de minúsculas y pueden incluir espacios y caracteres especiales. Sin embargo, los nombres de variable que incluyen caracteres especiales y espacios son difíciles de usar y deben evitarse. Para obtener más información, vea nombres de variable que incluyen caracteres especiales.

# Tipos de variables en PowerShell



Variables creadas por el usuario: el usuario crea y mantiene las variables. De forma predeterminada, las variables que se crean en la línea de comandos de PowerShell solo existen mientras la ventana de PowerShell está abierta. Cuando se cierra la ventana de PowerShell, se eliminan las variables. Para guardar una variable, deberán agregarla a su perfil de PowerShell.



Variables automáticas: estas almacenan el estado de PowerShell. Estas variables las crea PowerShell y cambia sus valores según sea necesario para mantener su precisión. Los usuarios no pueden cambiar el valor de estas variables.



Variables de preferencia: estas almacenan las preferencias de usuario para PowerShell. Estas variables las crea PowerShell y se rellenan con valores predeterminados. Los usuarios pueden cambiar sus valores.

# ¿Qué son las estructuras de control?

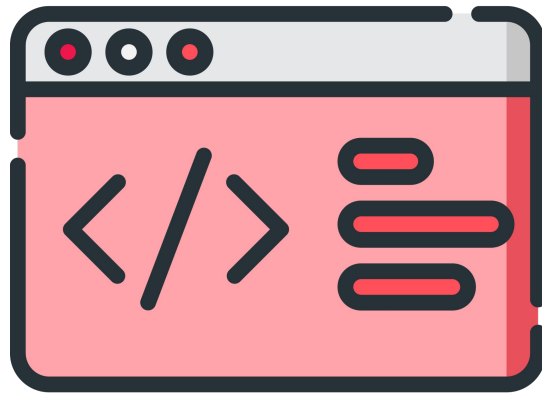
Las estructuras de control permiten modificar el flujo de ejecución de las instrucciones de un programa. Con ellas se puede:

- De acuerdo con una condición, ejecutar un grupo u otro de sentencias (If-Then-Else).
- De acuerdo con el valor de una variable, ejecutar un grupo u otro de sentencias (Switch-Case).
- Ejecutar un grupo de sentencias solo cuando se cumpla una condición (Do-While).
- Ejecutar un grupo de sentencias hasta que se cumpla una condición (Do-Until).
- Ejecutar un grupo de sentencias un número determinado de veces (For-Next).

# Tipos de estructuras de control

Todas las estructuras de control tienen un único punto de entrada. Las estructuras de control se pueden clasificar en: **secuenciales, iterativas y de control avanzadas**. Esta es una de las cosas que permiten que la programación se rija por los principios de la programación estructurada.

Los lenguajes de programación modernos tienen estructuras de control similares. Básicamente lo que varía entre las estructuras de control de los diferentes lenguajes es su sintaxis. Cada lenguaje tiene una sintaxis propia para expresar la estructura.

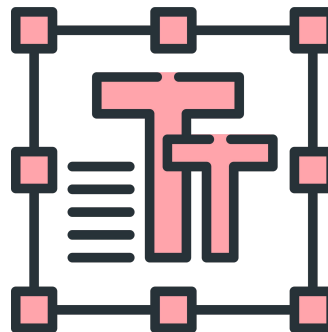


# ¿Qué es un script de PowerShell?

Un script es un archivo de texto sin formato que contiene uno o más comandos de PowerShell. Los scripts de PowerShell tienen una extensión de archivo **.ps1**.

Ejecutar un script es muy parecido a ejecutar un **cmdlet**. Pueden ejecutar scripts en el equipo o en una sesión remota en otro equipo.

Al escribir un script se guarda un comando para su uso posterior y facilita el uso compartido con otros usuarios. Lo más importante es que **permite ejecutar los comandos simplemente escribiendo la ruta de acceso del script y el nombre de archivo**. Los scripts pueden ser tan simples como un solo comando en un archivo o tan complejos como un programa completo.





# ¿Qué es una función en PowerShell?

En PowerShell, como en muchos lenguajes, una función es un conjunto de instrucciones a las que les damos un nombre. El principal interés de las funciones es que podemos llamarlas varias veces, sin tener que volver a escribir las instrucciones en cada llamada.

Si se escriben scripts o comandos únicos de una línea de PowerShell y encuentran que a menudo tienen que modificarlos para distintos escenarios, es bastante probable que estos sean buenos candidatos para convertirlos en una función que se pueda volver a usar.



DigitalHouse>