**GESTIÓN DE ALIAS, ARCHIVOS Y PIPELINE EN POWERSHELL**

**GESTIÓN DE ALIAS**

Los alias son una manera efectiva de abreviar los nombres de los comandos. Powershell viene con una buena cantidad de alias integrados. Se pueden consultar mediante el comando:

get-alias

Si se desea consultar si cierto comando tiene alias, se puede hacer mediante el mismo comando, con el parámetro -Definition. Por ejemplo:

get-alias -Definition "Get-Service"

... mostraría los alias correspondientes al comando Get-Service.

Para crear un alias nuevo se emplea el comando New-Alias. Por ejemplo:

new-alias -Name np -Value Notepad

... crea un alias llamado np para el comando Notepad (bloc de notas).

El resto de los comandos para alias se puede consultar con el comando:

help \*alias

Los alias definidos solamente son válidos para la sesión en la que se definan. Una vez se sale de sesión, dichos alias se pierden.

**GESTIÓN DE ARCHIVOS Y DIRECTORIOS**

En la filosofía de Powershell, los archivos y los directorios se consideran como objetos tipo **ITEM**. Por lo tanto, se gestionan con los siguientes comandos:

new-item

remove-item

get-item

get-childitem

move-item

rename-item

get-itemproperty

set-itemproperty

Los comandos finalizados en **Property** permiten gestionar propiedades de los archivos y directorios, tales como fechas de creación, acceso y modificación, y permisos de acceso.

**GESTIÓN DE DIREnewCTORIOS**

| **Operación** | **Comando** |
| --- | --- |
| Creación | new-item -itemtype directory -name nombre\_directorio |
| Borrado | remove-item -path nombre\_directorio |
| Renombrado | rename-item -path nombre\_directorio -newname nombre\_nuevo |
| Mover | move-item -path nombre\_directorio -destination nuevo\_sitio |
| Ingreso | cd nombre\_directorio |

Vale la pena recordar que . (punto) indica el directorio actual y .. (punto punto) indica el padre del directorio actual.

Para mostrar el directorio de trabajo se puede emplear el comando Get-Location, que tiene como alias pwd (igual que en Linux).

**GESTIÓN DE ARCHIVOS**

| **Operación** | **Comando** |
| --- | --- |
| Creación | new-item -itemtype file -name nombre\_arch [-value contenido] |
| Borrado | remove-item -path nombre\_archivo |
| Renombrado | rename-item -path nombre\_archivo -newname nombre\_nuevo |
| Mover | move-item -path nombre\_archivo -destination nuevo-sitio |

Los archivos de texto se pueden mostrar por consola con el comando Get-Content, que tiene como alias type.

**PROPIEDADES DE ARCHIVOS Y DIRECTORIOS**

Las propiedades de archivos y dirtectorios se pueden examinar con el comando Get-ItemProperty. Para Powershell los archivos y los directorios son *OBJETOS*, y sus miembros (propiedades y métodos, entre otros) se pueden listar mediante el comando get-member, de la siguiente manera:

get-itemproperty -path nombre\_arch | get-member

Algunas propiedades interesantes son: CreationTime, FullName, LastAccessTime, LastWriteTime, Attributes.

Por ejemplo, para consultar la fecha de creación de un archivo:

get-itemproperty -path nombre\_arch -name CreationTime

Para consultar los atributos (permisos) del archivo:

get-itemproperty -path archivo -name Attributes

Para fijar los atributos se emplea el comando Set-Itemproperty, por ejemplo:

set-itemproperty -path archivo -name Attributes -value "ReadOnly"

Algunos valores de permisos son ReadOnly, Hidden, System. Si se desea fijar varios al tiempo, se separan sus nombres con comas, y se encierra todo el conjunto entre comillas (como en el ejemplo).

**EL PIPELINE**

El pipeline (o canalización) permite conectar la salida de un comando con la entrada de otro. En el apartado anterior se vio un ejemplo de su uso:

Get-ItemProperty -path nombre\_archivo | get-member

El poder del pipeline consiste en poder alterar o filtrar la salida de un comando, para obtener la información que realmente se requiere. Por ejemplo, el comando:

get-process

... muestra los procesos organizados por orden alfabético según nombre de proceso. Si se deseara ordenar por ID de proceso, se puede emplear el pipeline de la siguiente manera:

get-process | Sort-Object -Property id