Contenido

[Directorios y archivos, crear y mover ,copiar: 2](#_Toc86158401)

[Información y modificación de archivos: 3](#_Toc86158402)

[Arrays, listas y csv 4](#_Toc86158403)

[Variables: 7](#_Toc86158404)

[Variables seguidas de un punto: 11](#_Toc86158405)

[Información y modificación de discos: 12](#_Toc86158406)

[Información del sistema y adaptadores de red: 13](#_Toc86158407)

[Discos y particiones 14](#_Toc86158408)

[Usuarios y grupos: 15](#_Toc86158409)

[Scripts: 16](#_Toc86158410)

# Directorios y archivos, crear y mover ,copiar:

1-comprobar el contenido de la carpeta de datos:

Get-ChildItem -Path $HOME -Force



2-crear carpetas; musikal, imaginal y textos:

New-Item -ItemTypedirectory -Pathdireccion, direccion si hayque hacer otra..



3-copiar un documento de texto en un dispositivo

Copy-Item -Pathdireccion -Destination donde le quieres



4-eliminar un archivo:

Remove-Item .\desktop\direccion



5-mover la carpeta

Move-Item -Pathdireccion del archivo -Destinationdireccion de destino



6-crear un documento de texto, con nombre luis.txt y que contenga 5 nombres:

New-Item -ItemType file -Name luis.txt -Valueluis,nano, jorge, valera



7-comprobar el contenido de un archivo:

Get-Content .\luis.txt



8-comprobar si existe un fichero:



9-renombrar un fichero:

Ren .\fichero.txt nombrenuevo.txt



# Información y modificación de archivos:

Mostrar información sobre archivos:

Get-childItem –Path $home

Crear archivos

New-Item -ItemType File -path $home -name Archivo1

Crear carpetas

New-Item -path $home -ItemTypeDirectory -name Caarpeta1

Mostrar archivos ocultos:

Get-childItem –Path $home –Hidden –Recurse

Obtener los descriptores de seguridad de todos los ficheros de una unidad:

Get-ChildItem D:\ -Recurse | %{Get-Acl $\_.FullNAme | Select-ObjectSddl}

Cambiar la fecha de un archivo:

Get-Item D:\Pruebas\Solvetic |%{$\_.CreationTime='10/01/2018 09:00:00'}

Listar los archivos abiertos recientemente

Get-ChildItem ([Environment]::GetFolderPath(“Recent”))

Obtener un listado de todos los ficheros exe:

Get-ChildItemc:\”\*.exe” –recurse | Select-ObjectFullName,Name

Almacenar contenido de un fichero:

Get-Process | Out-File c:\scripts\procesos.txt

Agregar la palabra fin a todos los ficheros txt

Add-Content –Path \*.txt –Value “FIN”

Ver el contenido de un fichero:

Get-Content fichero.txt

Ver la última línea de un fichero:

Get-Content .\fichero.txt –Tail 1

Comprimir un archivo:

Compress-Archive –LiteralPath c:\ejemplo.txt –CompressionLevelOptimal –DestinationPath C:\comprimido.zip

Encontrar la fecha de modificación de un archivo.

$lastModifiedDate = (Get-Item "C:\foo.tmp").LastWriteTime;

Ver los permisos:

Get-Acl –Path archivo

Establecer permisos:

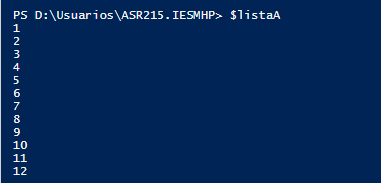
Set-Acl –Path archivo

# Arrays, listas y csv

Crea una listaA lista simple con los números del 1 al 12.



Muestra el tamaño de lista.



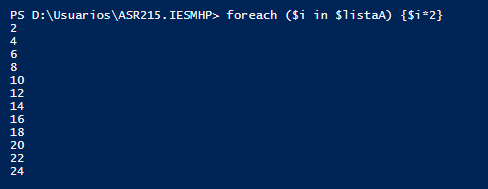
Lista el contenido de la lista.



Muestra el valor de la lista, de índice 2.



Recorre la lista y da cada valor de la lista multiplicado por 2.



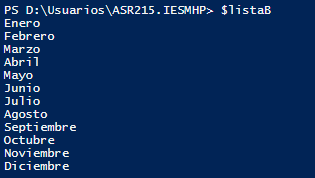
Crea una listaB lista simple con los meses del año.



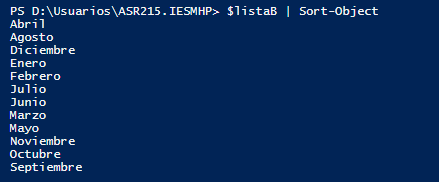
Muestra el tamaño de listaB.



Lista el contenido de la listaB.



Lista el contenido de la listaB ordenado alfabéticamente.



Recorre los elementos de la listaB con un foreach, y con un for.

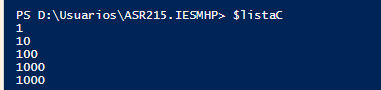
Crea la listaC con los siguientes números:1,10,100,1000,1000.



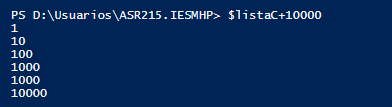
Muestra el tamaño de la listaC.



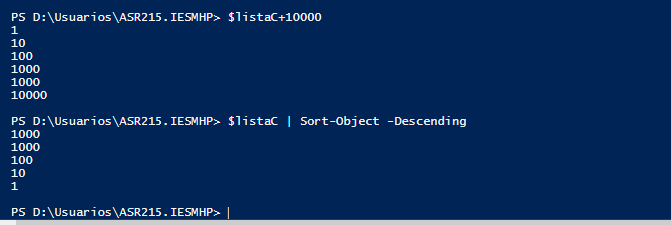
Lista el contenido de la listaC.



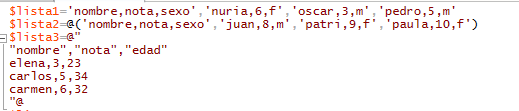
Añade el valor 10000 a la listaC.



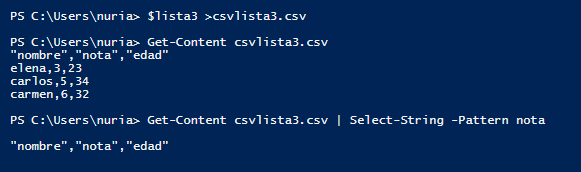
Lista el contenido de la listaC en orden descendente.



Define las siguientes listas:

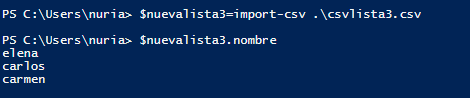


Guarda el contenido de lista3 en un fichero llamado csvlista3.csv.



Pasa el fichero csv anterior a una variable que llamarás $nuevalista3, usando el comando import-csv.

Lista por columnas los elementos de $nuevalista3



Otras maneras de listar columnas con foreach y con select-object

Define el hash $productos



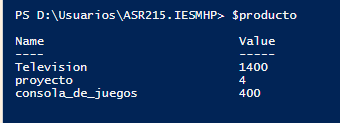
Con los siguientes datos:

| **Clave** | **Valor** |
| --- | --- |
| **Proyector** | **1600** |
| **Television** | **1400** |
| **Consola\_de\_juegos** | **400** |



Muestra de varias maneras el contenido del hash

Saca el listado sólo de los productos.



Saca el listado sólo de los valores.

Añade el producto router con el valor 150.



Indica el número de pares del hash $productos

Convierte la lista1 en un csv y lista por columnas su contenido.

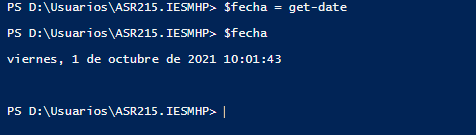
Convierte la lista2 en csv y lista por columnas su contenido.



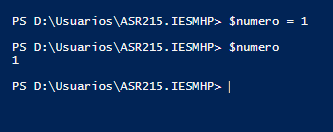
# Variables:

1. Prueba los siguientes ejemplos:

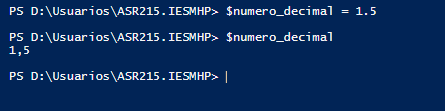
PS>$fecha = get-date



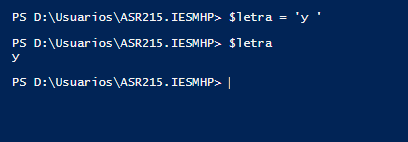
PS>$numero = 1



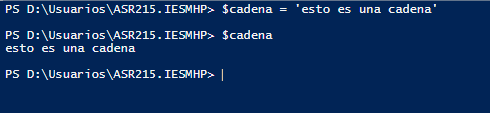
PS>$numero\_decimal = 1.5



PS>$letra = 'y '

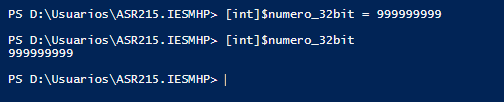


PS>$cadena = 'esto es una cadena'

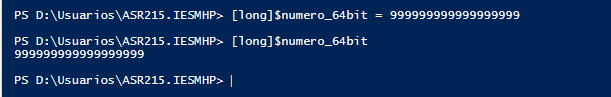


Puedes asignar eltipoconcreto de variable indicándolo entre corchetes. Estos son algunos:

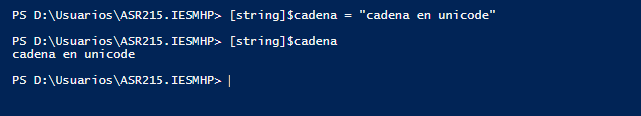
[int]$numero\_32bit = 999999999



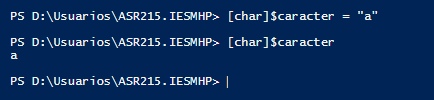
[long]$numero\_64bit = 999999999999999999



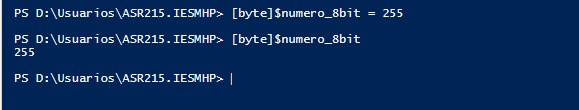
[string]$cadena = "cadena en unicode"



[char]$caracter = "a"



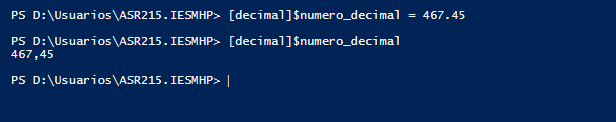
[byte]$numero\_8bit = 255



[bool]$verdad = $true



[decimal]$numero\_decimal = 467.45



Para mostrar el valor de una variable, simplemente escríbela y pulsa ENTER

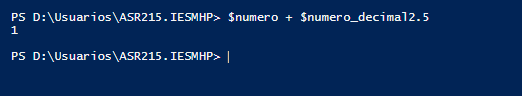
**PS>$numero1 PS>$cadenaesto es una cadena**

Puedes sumar dos o más números:

PS>1 + 1.5 2.5

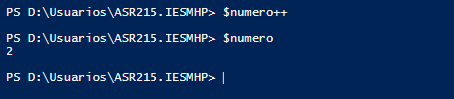
o sumar dos o más variables que contienen números:

PS>$numero + $numero\_decimal2.5



o incrementar directamente $numeroen una unidad:

PS>$numero++



PS>$numero2Puedes unir dos o más cadenas:

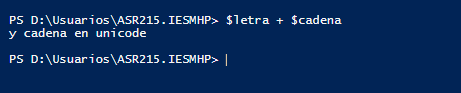
PS>'y ' + 'esto es una cadena'

y esto es una cadena

Observa que los espacios en blanco se tienen en cuenta.

Para unir dos o más variables que contienen cadenas:

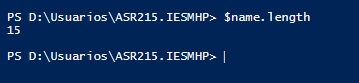
PS>$letra + $cadena



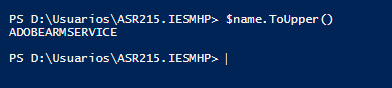
y esto es una cadena.

# Variables seguidas de un punto:

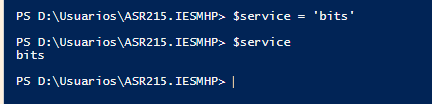
$name.lengthla longitud que ocupa ese archivo



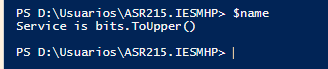
$name.ToUpper()la variable con todas las letras en mayusculas



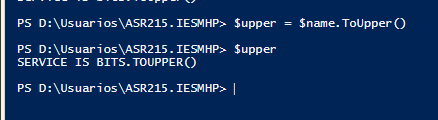
$service = 'bits'



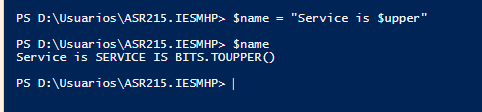
$name = "Serviceis $service.ToUpper()"



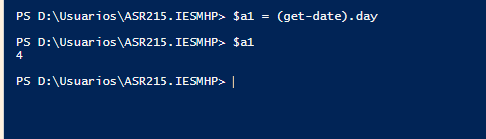
$upper = $name.ToUpper()todo en mayuscula



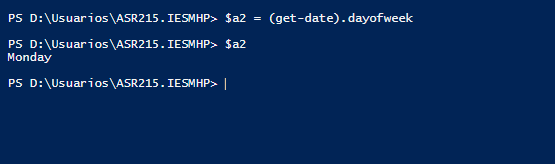
$name = "Serviceis $upper"te lo da en mayuscula



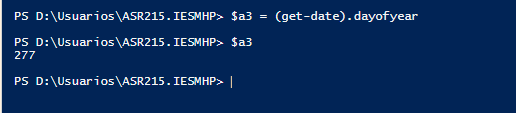
#$a1 = (get-date).dayde el comando que se le pone solo coge el dia que es



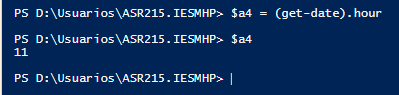
#$a2 = (get-date).dayofweekdia de la semana



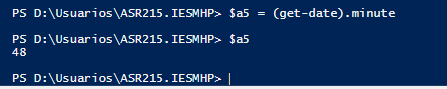
#$a3 = (get-date).dayofyear



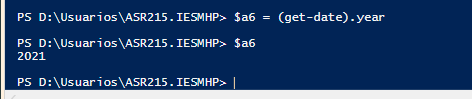
#$a4 = (get-date).hourque de ese comando solo saque la hora



#$a5 = (get-date).minutesolo te de los minutos



#$a6 = (get-date).year



# Información y modificación de discos:

Comprobar que reconoce los discos:



Comprobar las particiones:



Formatear un volumen:



Desfragmentar el disco:



Crear un punto de restauración:



# Información del sistema y adaptadores de red:

Ver la versión de powershell:

Get-host

Nombre del equipo:

hostname

Información de red:

Get-NetIpAddress

Información de router:

Get-netRoute

Cambiar el nombre del equipo:

Rename-computer “nombre del equipo”

Añadir un equipo a un directorio de trabajo:

Add-computer –workgroupName “nombre del grupo de trabajo”

Información de las tarjetas de red:

Get-netadapter

Renombrar la tarjeta de red:

Rename-NetAdapter -Name Ethernet0 -NewName RED\_LAN

Información de las direcciones del equipo:

Get-netIPadress

Información de las interfaces:

Get-netIPInterface

Quitar cualquier configuración de configuración ip del adaptador de red:

Remove-NetIPInterface –interfaceIndex 22 –confirm $false

Habilitar adaptador de red:

Enable-netAdaptader

Habilitar dhcp:

Set-NetIPInterface -InterfaceAliasWiFi -DhcpEnabled

# Discos y particiones

|  |  |
| --- | --- |
| Información sobre los discos físicos | Get-disk |
| Información lógica sobre particiones | Get-partition |
| Información de las particiones | Get-volume |
| Crear una nueva partición | New-Partition -DiskNumber 1 -Size 12GB -DriveLetter X  Con este parámetro indicamos lo siguiente:   * **New-Partition:** cmdlet que nos permite crear la partición. * **-DiskNumber:** Indicamos el número de disco donde crearemos la partición. * **-Size:** Indicamos el tamaño de la partición. * **-DriveLetter:** Asignamos un número de letra a la unidad.   En caso de querer particionar todo el espacio podemos usar el parámetro **-UseMaximumSize** y si deseamos que el sistema añada automáticamente la letra de la unidad usaremos el parámetro **-AssignDriveLetter** |
| Borrar la partición | Remove-Partition -DriveLetter E |
| Formatear (borrar) un disco | Clear-Disk 1 -RemoveData |
| Inicializar un disco | Initialize-Disk -Number 1 -PartitionStyle GPT |
| Formatear | Format-volume -DriveLetter E -FileSystem NTFS |
| Redimensionarparticion | Resize-partition |

# Usuarios y grupos:

|  |  |
| --- | --- |
| Ver los usuarios locales | **Get-localuser** |
| Ver los grupos locales | **Get-localgroup** |
| Crear un usuario | **New-LocalUser**“nombre\_usuario”**-Password** “contraseña1” –**Fullname**“Nombre completo”**-Description**“Descripción de la cuenta” |
|  | **$Password** = Read-Host –AsSecureString  **New-LocalUser** “nombre\_usuario”**-Password**$Password –**FullName** “Nombre completo”**-Description** “Descripción de la cuenta” |
|  | **New-LocalUser** “nombre\_usuario”**-Password**$Password –**FullName** “Nombre completo”**-Description** “Descripción de la cuenta” |
| Cambiar la contraseña | $pass2=ConvertTo-SecureString "11234aaaf" -asplaintext -force  Set-LocalUser -Name usuario -Password $pass2 |
| Crear un usuario sin contraseña | New-LocalUser -Name "nombre\_usuario" -Description "Descripcion de la cuenta" -NoPassword |
| Eliminar un usuario | Remove-LocalUser -Name "usuario" |
| Añadir un usuario a un grupo local | **Add-LocalGroupMember -Group** “Administradores” **-Member** “usuario\_creado” |
| Crear un grupo local | #Es necesario abrir PowerShell como administrador  New-LocalGroup grupo |
|  | Add-LocalGroupMember -Member usuario -Group administradores  Get-LocalGroupMember administradores |
| Eliminar usuarios de grupos | Remove-LocalGroupMember-Memberusuario-Groupgrupo |
| Eliminar grupos | Remove-LocalGroupgrupo |

# Scripts: