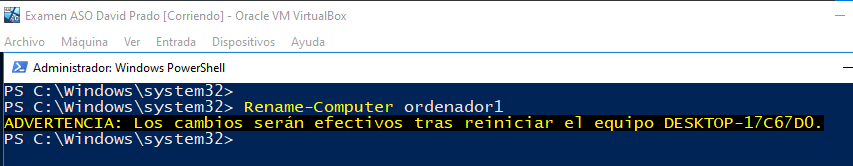
|  |  |
| --- | --- |
| Administración de Sistemas Operativos | |
| **Aplicación de scripting en SO**  **PowerShell Examen** | |
| Fecha 27 Octubre 2021 |  |
| El trabajo ha de ser perfectamente legible. Haz los recortes necesarios para justificar el trabajo, primero del script o comando y luego de su salida. Guárdalo en un pdf con encabezado tu nombre y fecha y súbelo a classroom ASO examen\_bloque2. (no cumplir el formato resta 1 del total obtenido)  Se valora que todos en los ejercicios se incluyán comentarios explicando el flujo del programa en su caso y/o se valorará la claridad y precisión así como la estética de los mismos. | |

1. Tienes un equipo con 2 tarjetas de red (Puente/NAT y red interna1), y 2 discos duros, el primero con Windows 10 instalado.

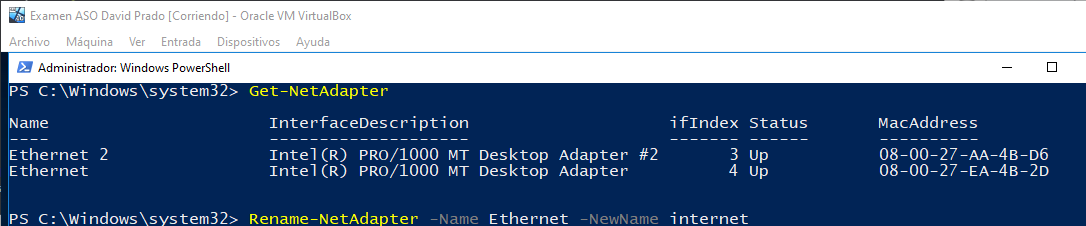
Prepara el equipo mediante un script de comandos **PowerShell** de la siguiente manera:

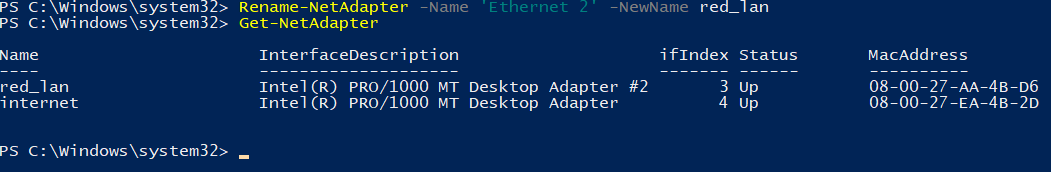
* Nombre equipo windows10: ordenador1 (0,2 puntos)



* Nombre de la interface de la tarjeta NAT: internet (0,2 puntos)
* Nombre de la interface de la tarjeta en red interna: red\_LAN (0,2 puntos)

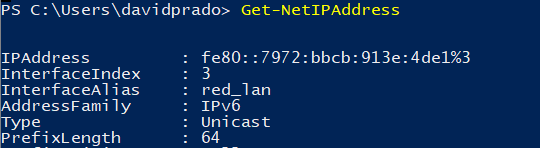
Mirando las MACs de las tarjetas de red puedo determinar cual es la interna y la externa

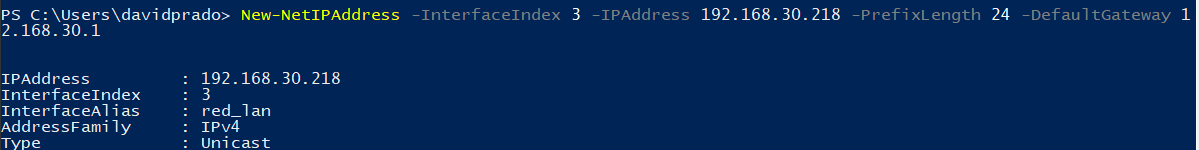




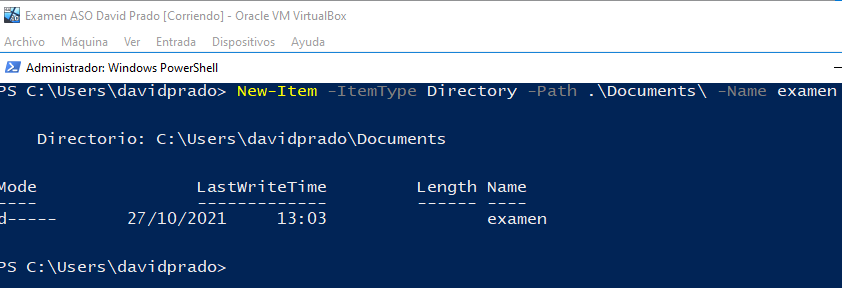
* Ponle la IP fija a la tarjeta red\_LAN: 192.168.30.xx, donde xx es tu nº de equipo. (0,2puntos)

**Con el comando get-netipaddress puedo saber el numero index de cada tarjeta de red**

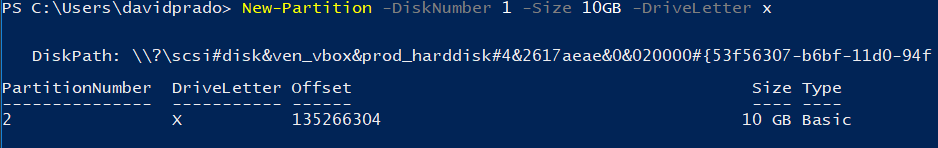


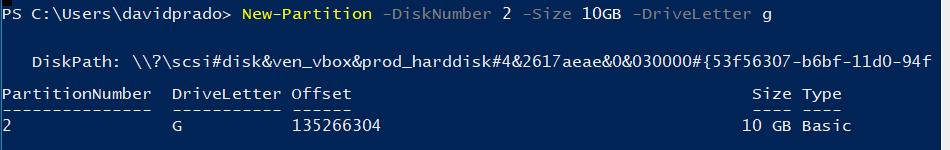


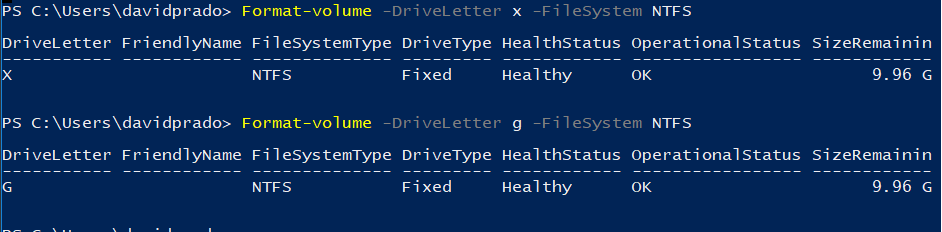
* Crea la carpeta examen en el directorio Documentos. (0,2 puntos)



* Crea en el disco 2, 3 particiones con formato NTFS, de tamaño 20 GB cada uno. (0,2 puntos)

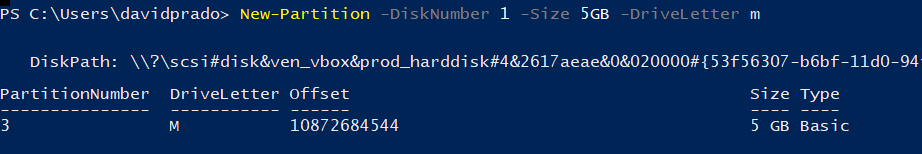


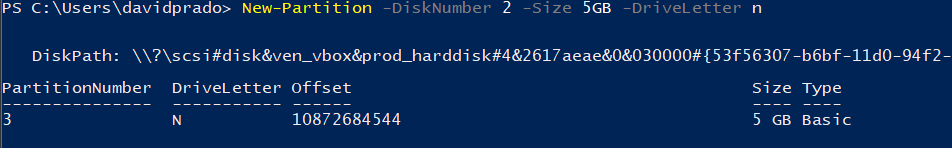


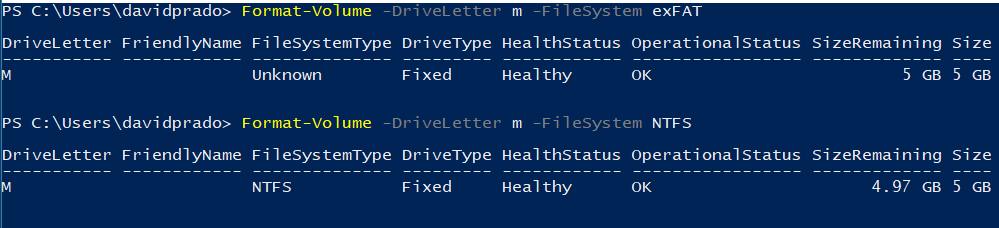


* Crea en el disco 1, dos particiones de 30 GB, una con formato exfat y otra NTFS con letras de unidad M y otra N. (0,2 puntos)

**Yo las he creado en el disco 2 y 3 ya que en el dsico 1 no me dejaba a pesar de que tenia 30 gigas libres**







1. Si tenemos en cuenta la siguiente clasificación de ordenadores de sobremesa según sus usos:

* *Aplicaciones de oficina*: la mayoría, tanto en el ámbito doméstico como laboral, se utilizan principalmente para ejecutar aplicaciones de oficina, como tratamiento de textos, navegar o correo electrónico. [procesador Corie i3](https://www.quonty.com/procesadores/) y una **memoria RAM 4G**
* *Videojuegos*: los ordenadores de sobremesa para jugar deben ser potentes en todos los aspectos, procesador, tarjeta gráfica, tarjeta de sonido, disco duro y pantalla. Disponen del hardware más potente y están optimizados para obtener el máximo rendimiento del procesador y la tarjeta gráfica. una buena [tarjeta gráfica](https://www.quonty.com/tarjetas-graficas/) y una [memoria RAM](https://www.quonty.com/memorias-ram-para-gaming/) de al menos **8GB** a **DDR3,** además de un [buen sistema de refrigeración](https://www.quonty.com/blog/refrigeracion-liquida-vs-refrigeracion-por-aire-para-cpu-cual-elegir/) que evita el sobrecalentamiento de tu equipo.
* *Multimedia*: para actividades multimedia conviene priorizar un disco duro grande y una elevada conectividad, con USB, HDMI, ranura para tarjetas de memoria, Ethernet, Wi-Fi, Bluetooth, DVI y grabadora DVD. PC sobremesa con un [buen procesador](https://www.quonty.com/procesadores/), una**gran memoria RAM de al menos 8GB** pero si tu presupuesto te lo permite, mejor de **16GB** o **32GB** y dos o cuatro módulos y una tarjeta gráfica muy potente para soportar las exigencias propias del uso que vas a dar a tu equipo

Se pide crear un script que me muestre, indicando lo q se va a visionar, la siguiente información concreta de tu equipo:

* Tamaño del disco1
* Tipo de disco duro del disco1
* Marca o fabricante del procesador
* Velocidad de trabajo
* Tamaño de la memoria RAM máxima que soporta
* Tamaño de memoria RAM efectiva
* Modelo/fabricante de placa base wmic baseboard
* Tipo de tarjeta gráfica wmic path win32\_VideoController
* Tipo de tarjeta de sonido
* Tipo de refrigeración

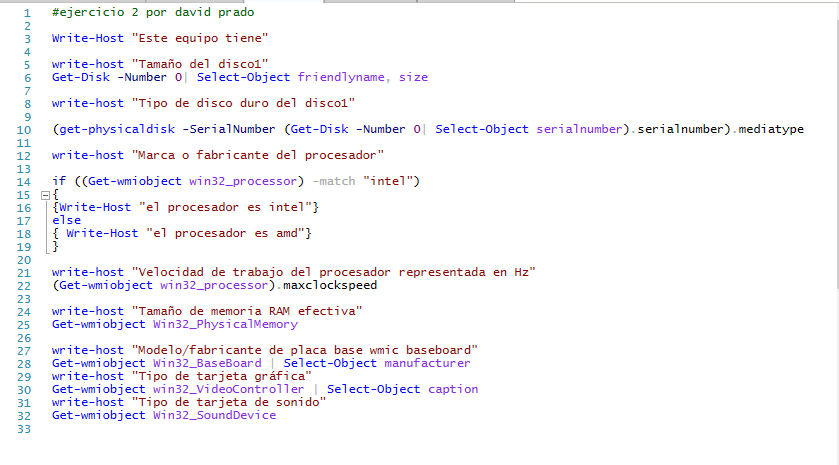
Get-WmiObject Win32\_Fan

Get-WmiObject Win32\_HeatPipe

Get-WmiObject Win32\_Refrigeration

Get-WmiObject Win32\_TemperatureProbe

(Concreta con información relevante que tú valores como técnico) Sin ninguna línea irrelevante (serán penalizadas), en función de esa información y mande la información a un archivo txt de nombre mi\_equipo.txt y te indique para lo que mejor esté orientado (oficina, multimedia o juegos). (2 puntos comandos exactos+ testeo y script)



1. Se pide un script, que a partir de una lista (puede ser un csv/ o un txt) con el nombre, apellido1, edad (introduce 2 mayores de edad y 3 menores de edad) y sexo (hombre/mujer) de 5 personas, generes:

* Una carpeta para cada usuario, llamada nombre.Apellido1
* Cree un usuario local para cada uno, con la contraseña pepe@2 (1,2 puntos)

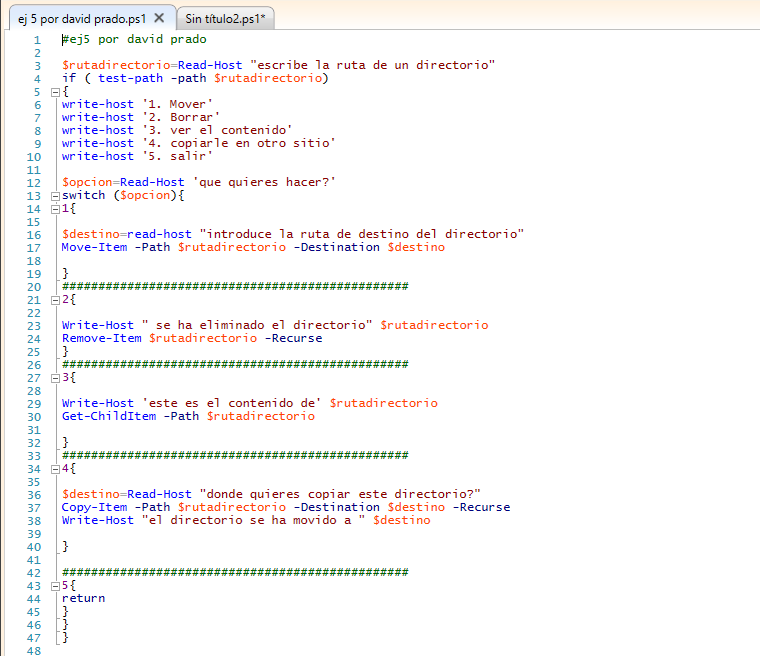
Y por otro lado genera otro tipo menú, que te haga lo siguiente:

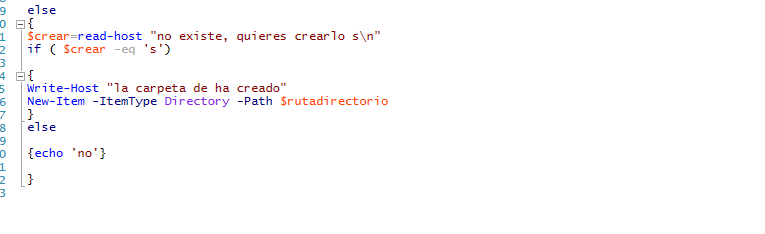
Listar sólo los nombres de la lista.

* Listar los mayores de 18 años.
* Listar la cantidad de menores.
* Salir.

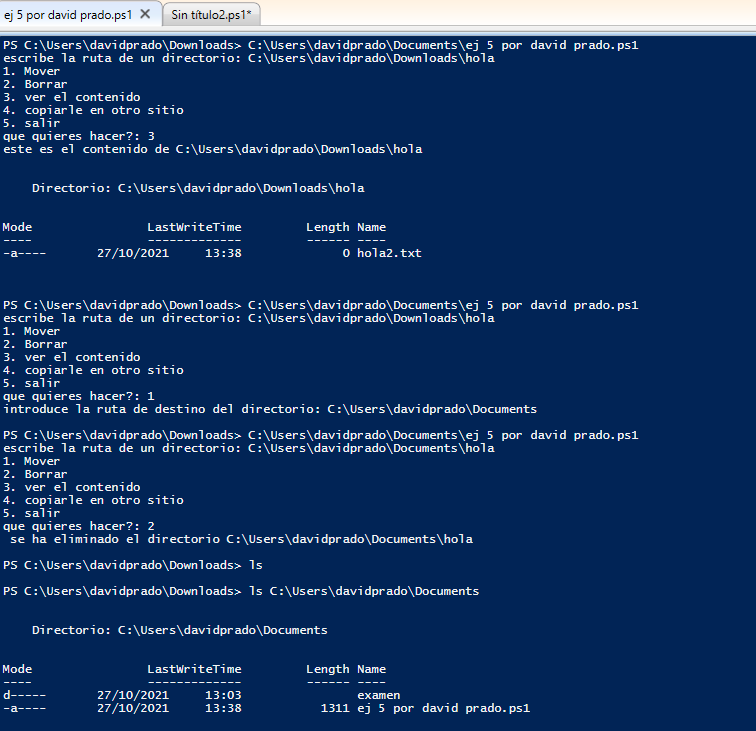
(1.2 puntos)

1. Se pide un script que me convierta una imagen .bmp, usando parámetros posicionales, en .jpg. y luego me liste todas las imágenes jpg, de la carpeta imágenes del equipo. (1 punto)
2. Se pide un script que te dé distintas opciones a realizar sobre directorios:

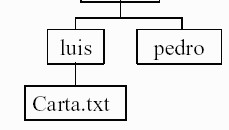
Crear, mover, borrar, ver el contenido (distinguiendo entre directorios y ficheros), copiar un directorio, a partir de una ruta y el nombre de una carpeta, primero habrá de testear si existe. Cada vez que pulse una opción tenemos que introducir los parámetros necesarios de esa opción, (1,4 puntos)  




**Para hacer la muestra del script he creado el directorio hola dentro de descargas**



1. Crea la siguiente estructura de directorios en tu directorio de trabajo:

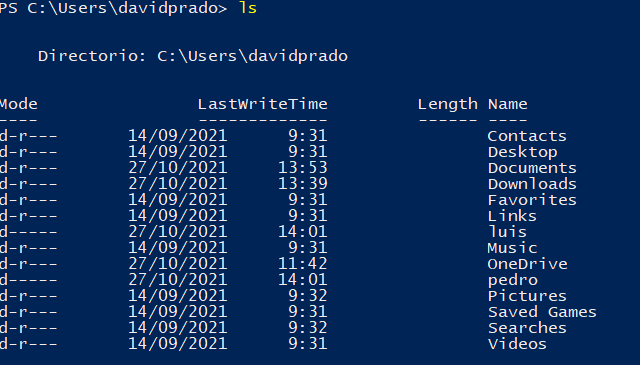


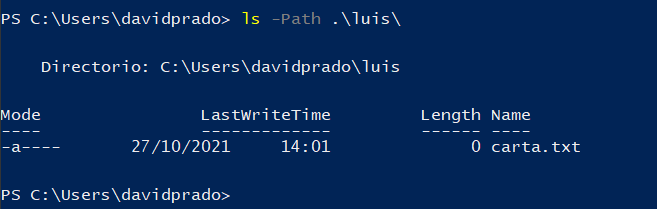
Donde Luis y Pedro son directorios y Carta.txt un documento de texto de la carpeta Luis. (0,3 puntos)



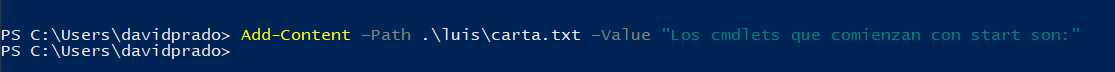






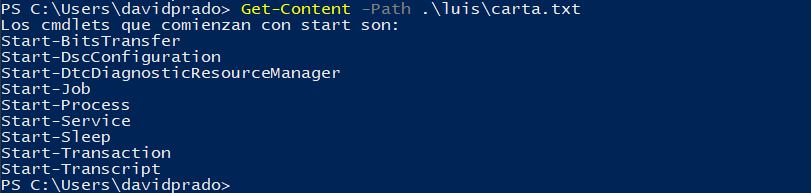


* Crea la siguiente línea al documento carta.txt: “Los cmdlets que comienzan con start son:”(0,2 puntos)

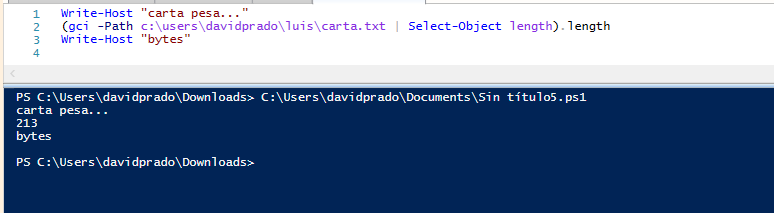


* Añade el listado con los cmdlets que comienzan con start al documento carta.txt. (0,2 puntos)





* Obtén el tamaño del documento carta.txt (0,2 puntos)



TEST (1 punto: las bien +0.1, las mal -0.05. Sombrea en amarillo la opción elegida)

1. Para completar un cmdlet en la siguiente línea usas:

a) F7

b) Tabulador

c) -(guión)

d) `(acento grave)

2. Powershell está incorporada en todas las versiones de Windows:

a) Sólo en los Windows server

b) Cada versión de Windows tiene su versión.

c) Todos los Windows tiene la versión 5 o superior de powershell.

d) Las versiones de powershell no son actualizables.

3. Cuando intentas ejecutar un script por primera vez con PowerShell:

a) has de ejecutarlo como administrador y habilitar la ejecución de scripts con set-executionPolicy Unrestricted.

b) Se ejecuta como cualquier script.

c) Debes de importar el modulo de Active Directory.

d) Debes de habilitarlo con get-member.

4. Para crear el fichero texto1.txt en el directorio D:\Datos\textos. Cual no es correcto:

a) New-Item “D:\Datos\textos” texto1.txt file

b) Write-host “ “ > ; D:\Datos\Textos\texto1.txt

c) New-Item”–ItemType file –path “D:\Datos\textos

d) New-Item file “D:\Datos\textos

5. Para crear la variable mareo con el valor esto es lo que hay, cuál de las siguientes no es correcta:

a) $mareo=”esto es lo que hay”

b) New-variable –name mareo –value “esto es lo que hay”

c) Set-variable –name mareo –value “esto es lo que hay”

d) Get-variable –name mareo –value “esto es lo que hay”

6. El identificado del primer parámetro posicional es:

a) %1

b) Args[0]

c) $\_

d) $?

7. El primer valor del array $array es:

a) $array[0]

b) $array[1]

c) Args[0]

d) $1

8. Para obtener solamente la lista únicamente de las propiedades del objeto get-childitem, escribimos:

1. PS>get-childitem | get-member
2. PS>get-childitem –property
3. PS>get-childitem | get-member | where-object{$\_.membertype –eq “property”}
4. PS>get-childitem | where-object{$\_.membertype –eq “property”}

9. La expresión get-service | where-object{$.estatus –eq “stopped”}

1. Lista los servicios que están parados.
2. Lista los procesos que contengan la palabra stopped.
3. Lista los procesos que están parados.
4. Lista los procesos que no están parados.

10. La expresión get-childitem | where-object{$\_.length –ge 1000 }

1. Muestra los ficheros que tienen 1000MB o más del directorio actual.
2. Muestra los ficheros que tienen 1000MB o menos del directorio actual.
3. **Muestra los ficheros que tienen 1000 bytes o más del directorio actual.**
4. Muestra los ficheros que tienen 1000 Bytes o más del directorio raíz.