1. Utilizar Get-member para saber que propiedades podemos consultar de los objetos devueltos por Get-ChildItem sobre el directorio actual. Observa como tenemos acceso a propiedades diferentes según si el item representa un archivo o un directorio.

Get-ChildItem | Get-Member

```
PS C:\Users\Administrador.WIN-VOLV1BK8OEB> Get-ChildItem | Get-Member


   TypeName: System.IO.DirectoryInfo

Name                      MemberType     Definition
----                      ----------     ----------
LinkType                  CodeProperty   System.String LinkType{get=GetLinkType;}
Mode                      CodeProperty   System.String Mode{get=Mode;}
Target                    CodeProperty   System.Collections.Generic.IEnumerable`1[[System.String, mscorlib, Version=4.0.0.0, Cultu...
Create                    Method         void Create(), void Create(System.Security.AccessControl.DirectorySecurity directorySecur...
CreateObjRef              Method         System.Runtime.Remoting.ObjRef CreateObjRef(type requestedType)
CreateSubdirectory        Method         System.IO.DirectoryInfo CreateSubdirectory(string path), System.IO.DirectoryInfo CreateSu...
Delete                    Method         void Delete(), void Delete(bool recursive)
EnumerateDirectories      Method         System.Collections.Generic.IEnumerable[System.IO.DirectoryInfo] EnumerateDirectories(), S...
EnumerateFiles            Method         System.Collections.Generic.IEnumerable[System.IO.FileInfo] EnumerateFiles(), System.Colle...
```

2. Utilizar una propiedad de las consultadas en el apartado anterior y el cmdlet Where-Objet para obtener todas las .dll del directorio Windows y todos sus subdirectorios que tengan un tamaño mayor de 20MB. Si hay algún error por permisos de acceso se debe omitir.

Get-ChildItem | Where-Object {$_.Length -gt 20000*1024}

```
PS C:\Windows\System32> Get-ChildItem | Where-Object {$_.Length -gt 20000*1024}


    Directorio: C:\Windows\System32


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        16/07/2016     15:19       22571520 edgehtml.dll
-a----        16/07/2016     15:18       26217472 imageres.dll
-a----        16/07/2016     15:19       23681536 mshtml.dll
-a----        16/07/2016     15:18       22219328 shell32.dll
-a----        16/07/2016     15:20       32693432 WindowsCodecsRaw.dll
```

3. De los objetos anteriores devueltos sólo nos interesa conocer el nombre, el tamaño y la ruta completa.

Get-ChildItem | Foreach {$_.name +" "+ $_.Directory +" "+ $_.Length}

```
XboxGipRadioManager.dll C:\Windows\System32 65536
xcopy.exe C:\Windows\System32 47616
XInput1_4.dll C:\Windows\System32 37888
XInput9_1_0.dll C:\Windows\System32 10752
XInputUap.dll C:\Windows\System32 46592
xmlfilter.dll C:\Windows\System32 67072
xmllite.dll C:\Windows\System32 214360
xmlprovi.dll C:\Windows\System32 22016
xolehlp.dll C:\Windows\System32 62976
XpsDocumentTargetPrint.dll C:\Windows\System32 352256
XpsGdiConverter.dll C:\Windows\System32 532480
XpsPrint.dll C:\Windows\System32 1699840
XpsRasterService.dll C:\Windows\System32 260608
```

4. De los objetos anteriores devueltos sólo nos interesa ver el tamaño en MB
Get-ChildItem | foreach {$_.Name+""+$_.Length/1MB+" MB"}

```
xmlprovi.dll0.02099609375 MB
xolehlp.dll0.06005859375 MB
XpsDocumentTargetPrint.dll0.3359375 MB
XpsGdiConverter.dll0.5078125 MB
XpsPrint.dll1.62109375 MB
XpsRasterService.dll0.24853515625 MB
xpsservices.dll2.96630859375 MB
xwizard.dtd0.00382804870605469 MB
xwizard.exe0.060546875 MB
xwizards.dll0.4287109375 MB
xwreg.dll0.11279296875 MB
```

5. Las aplicaciones instaladas en Windows aparecen en la clave del registro HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall. Vamos a crear una unidad para manejarla más fácilmente y realizar algunas operaciones con las aplicaciones instaladas. Ejecuta **New-PSDrive -Name aplicaciones -PSProvider Registry -Root HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall**

```
PS C:\Windows\System32> New-PSDrive -Name aplicaciones -PSProvider Registry -Root HKLM:\Softwa


Name           Used (GB)     Free (GB) Provider      Root                                               Cu
                                                                                                        rr
                                                                                                        en
                                                                                                        tL
                                                                                                        oc
                                                                                                        at
                                                                                                        io
                                                                                                         n
----           ---------     --------- --------      ----                                               --
aplicac...                             Registry      HKEY_LOCAL_MACHINE\Software\Micr...
```

6. Ahora ejecutar get-childitem aplicaciones: . Observa las claves y entradas. Fíjate que hay una entrada “Display Name ” que es una propiedad del item (una entrada de una subclave)
Get-ChildItem aplicaciones:\ | where-object {$_.GetValue("DisplayName")}

```
PS C:\Windows\System32>
Get-ChildItem aplicaciones:\ | where-object {$_.GetValue("DisplayName")}


    Hive: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall

Name                           Property
----                           --------
FileZilla Server               UninstallString : "C:\Program Files\FileZilla
                               Server\Uninstall.exe"
                               InstallLocation : C:\Program Files\FileZilla Server
                               DisplayName     : FileZilla Server 1.2.0
                               DisplayIcon     : C:\Program Files\FileZilla
```

7. Utilizar get-member para ver las propiedades y métodos a los que tienes acceso en los ítems de esta unidad. Observa cómo hay un método GetValue, si le pasamos el nombre de una entrada y nos devuelve su valor.
Get-ChildItem aplicaciones:\ | Get-Member

```
PS C:\Windows\System32>
Get-ChildItem aplicaciones:\ | Get-Member


   TypeName: Microsoft.Win32.RegistryKey

Name                      MemberType   Definition
----                      ----------   ----------
Close                     Method       void Close()
CreateObjRef              Method       System.Runtime.Remoting.ObjRef CreateObjRef(type ...
CreateSubKey              Method       Microsoft.Win32.RegistryKey CreateSubKey(string s...
DeleteSubKey              Method       void DeleteSubKey(string subkey), void DeleteSubK...
DeleteSubKeyTree          Method       void DeleteSubKeyTree(string subkey), void Delete...
DeleteValue               Method       void DeleteValue(string name), void DeleteValue(s...
Dispose                   Method       void Dispose(), void IDisposable.Dispose()
Equals                    Method       bool Equals(System.Object obj)
```

8. Vamos a obtener para cada aplicación instalada el valor de su entrada “DisplayName”. Ejecuta: Get-ChildItem -Path aplicaciones: | ForEachObject -Process { $_.GetValue("DisplayName") } Observa como en este caso no podemos utilizar select-object porque no es una propiedad, es una invocación a un método y por tanto necesitamos un bloque de script.

Get-ChildItem -Path aplicaciones: | ForEach-Object -Process { $_.GetValue("DisplayName")}

```
Get-ChildItem -Path aplicaciones: | ForEach-Object -Process { $_.GetValue("DisplayName") }

FileZilla Server 1.2.0
Mozilla Firefox 60.9.0 ESR (x64 es-ES)
Mozilla Maintenance Service
Oracle VM VirtualBox Guest Additions 6.1.26
PuTTY release 0.76 (64-bit)
```

9. Modificar el comando anterior para obtener sólo la aplicación cuyo display name sea Oracle VM VirtualBox Guest Additions 5.0.20
Get-ChildItem -Path aplicaciones: | Where-Object -FilterScript { $_.GetValue("DisplayName") -eq "Oracle VM VirtualBox Guest Additions 5.0.20"}

```
PS C:\Windows\system32> Get-ChildItem -Path aplicaciones: | Where-Object -FilterScript {$_.GetValue("DisplayName") -eq "Oracle VM VirtualBox Guest Additior


    Hive: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

Name                           Property
----                           --------
Oracle VM VirtualBox Guest     DisplayName     : Oracle VM VirtualBox Guest Additions 6.1.26
Additions                      UninstallString : C:\Program Files\Oracle\VirtualBox Guest Additions\uninst.exe
                               DisplayVersion  : 6.1.26.0
                               URLInfoAbout    : http://www.virtualbox.org
                               Publisher       : Oracle Corporation


PS C:\Windows\system32>
```

10. Sabiendo que para cada aplicación instalada existe una propiedad llamada UninstallString que contiene la ruta absoluta al ejecutable de desinstalación. ¿Cómo desinstalarlas todas las aplicaciones que contienen vlc en su nombre?
Get-AppxPackage *vlc* | Remove-AppxPackage

```
PS C:\Windows\System32> Get-AppxPackage *vlc* | Remove-AppxPackage



PS C:\Windows\System32>
```

12. Buscar en todas las claves en HKCU:\Software que no tengan más de una subclave y que tengan exactamente 4 entradas.
Get-ChildItem -Path HKCU:\Software -Recurse | Where-Object -FilterScript {($_.SubKeyCount -le 1) -and ($_.ValueCount -eq 4) }

```
PS C:\Windows\System32> Get-ChildItem -Path HKCU:\Software -Recurse | Where-Object -FilterScri



    Hive: HKEY_CURRENT_USER\Software\Microsoft

Name                           Property
----                           --------
Command Processor              CompletionChar     : 9
                               DefaultColor       : 0
                               EnableExtensions   : 1
                               PathCompletionChar : 9
```

13. Obtener todos los procesos cuyo id sea mayor que 100 y menor que 400.
Get-Process | Where-Object {($_.Id -ge 100) -and ($_.Id -le 400)}

```
PS C:\Windows\System32> Get-Process | Where-Object {($_.Id -ge 100) -and ($_.Id -le 400)}



Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    225      10     1732       4160       0,28     364   0 csrss
   1010      60    58128     109920       1,75     332   1 SearchUI
     51       2      380       1180       0,14     288   0 smss
    440      34    11868      17456       0,33     308   0 svchost
    175      10     1992       7412       0,09     384   0 VBoxService
```

14. Recuperar todos los servicios que tengan dependencias de otros servicios. Bien porque necesite otros servicios para funcionar o porque otros servicios dependan de ellos. Consulta las propiedades necesarias con get-member.
Get-Service -DependentServices

```
PS C:\Windows\System32> Get-Service -DependentServices


Status   Name               DisplayName
------   ----               -----------
Stopped  applockerfltr      Controlador de filtro Smartlocker
Stopped  Audiosrv           Audio de Windows
Running  WdNisSvc           Servicio de inspección de red de Wi...
Running  WdNisDrv           Controlador del Sistema de inspecci...
Stopped  SharedAccess       Conexión compartida a Internet (ICS)
Stopped  RemoteAccess       Enrutamiento y acceso remoto
Stopped  PolicyAgent        Agente de directiva IPsec
Stopped  NcaSvc             Asistente para la conectividad de red
Running  MpsSvc             Firewall de Windows
Stopped  IKEEXT             Módulos de creación de claves de IP...
```

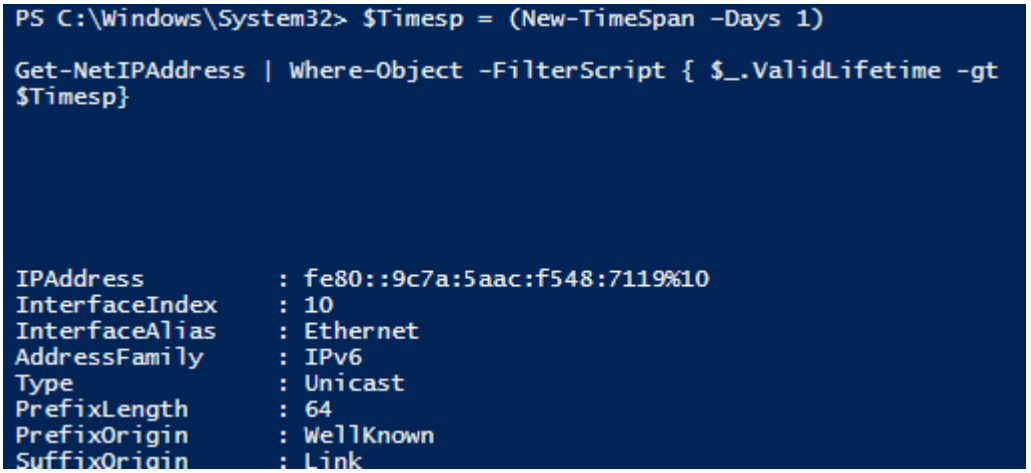
15. Repetir el apartado anterior pero mostrando para los servicios encontrados en formato de tabla las propiedades estado, nombre, servicios requeridos, servicios dependientes.
Get-Service -Name * -DependentServices | Get-Service -RequiredServices | Format-Table

```
PS C:\Windows\System32> Get-Service -Name * -DependentServices | Get-Service -RequiredService


Status   Name               DisplayName
------   ----               -----------
Stopped  AppIDSvc           Identidad de aplicación
Stopped  AppID              Controlador de AppId
Running  FltMgr             FltMgr
Stopped  AudioEndpointBu... Compilador de extremo de audio de W...
Running  RpcSs              Llamada a procedimiento remoto (RPC)
Running  WdNisDrv           Controlador del Sistema de inspecci...
```

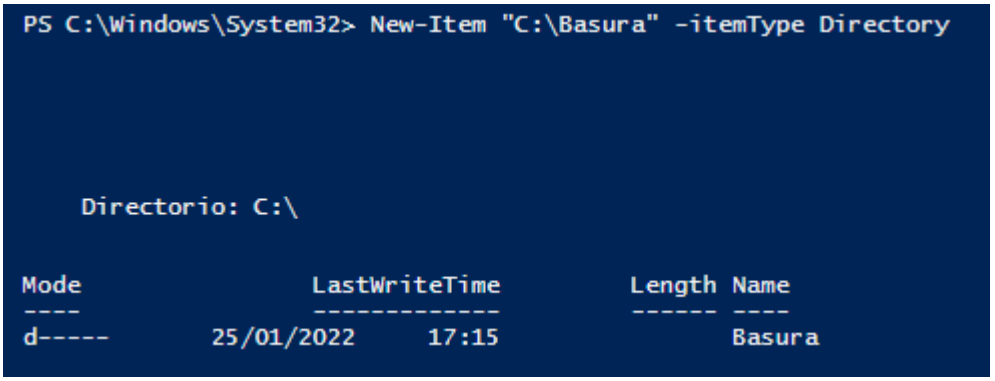
16. Reiniciar todos los servicios que se estén ejecutando cuyo nombre comience por i y que permitan reinicio. Añade –passThru para poder ver por pantalla los servicios reiniciados.
Get-Service -Name i* | Where-Object -FilterScript {$_.CanStop} | Restart-Service -PassThru

```
PS C:\Windows\System32> Get-Service -Name i* | Where-Object -FilterScript {$_.CanStop} | Resta


Status   Name               DisplayName
------   ----               -----------
Running  iphlpsvc           Aplicación auxiliar IP
```

17. Para consultar las ip con un tiempo de vida menor que un día. Ejecuta
$Timesp = (New-TimeSpan –Days 1)
Get-NetIPAddress | Where-Object -FilterScript { $_.ValidLifetime -Lt $Timesp}
Si usas get-member con Get-NetIPAddress puedes ver que ValidLifeTime es una propiedad de tipo DateTime no un número, por eso utilizamos NewTimeSpan para crear un objeto del tipo adecuado



18. Crear un directorio en C:\ llamado Basura. Siempre se trabajará en este directorio
New-Item "C:\Basura" -itemType Directory



19. Desarrolla un script en PowerShell llamado cambiar.ps1 que reciba un directorio, una extensión de archivo y una extensión nueva de manera que para todos los archivos con la extensión dada cambie la extensión la nueva proporcionada.
$ruta = Read-host "Introduce una ruta"

```
$ruta = Read-Host "Introduce una ruta"
$archivo = Read-Host "Introduce extensión de archivo"
$extensionNueva = Read-Host "Introduce extensión nueva"
cd $ruta
Get-ChildItem *.$archivo | rename-item -newname { [io.path]::ChangeExtension($_.name, "$extensionnueva") }
Get-ChildItem $ruta
```

```powershell
$ruta = Read-Host "Introduce una ruta"
$archivo = Read-Host "Introduce extensión de archivo"
$extensionNueva = Read-Host "Introduce extensión nueva"
cd $ruta
Get-ChildItem *.$archivo | rename-item -newname { [io.path]::ChangeExtension($_.name, "$extensionnueva") }
Get-ChildItem $ruta
```

```
PS C:\Users\Administrador\Documents> C:\Users\Administrador.WIN-V0LV1BK8OEB\Documents\copia.ps1
Introduce una ruta: C:\Users\Administrador\Documents
Introduce extensión de archivo: txt
Introduce extensión nueva: jpg


    Directorio: C:\Users\Administrador\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        26/01/2022      8:38                Nueva carpeta
-a----        26/01/2022      9:14              0 hola.jpg
```