

# SOC (Security Operations Center)

## Módulo 1



## Sistema Operativo de Windows Ejercicios

Sheila Fernández Cisneros – 17/05/2024

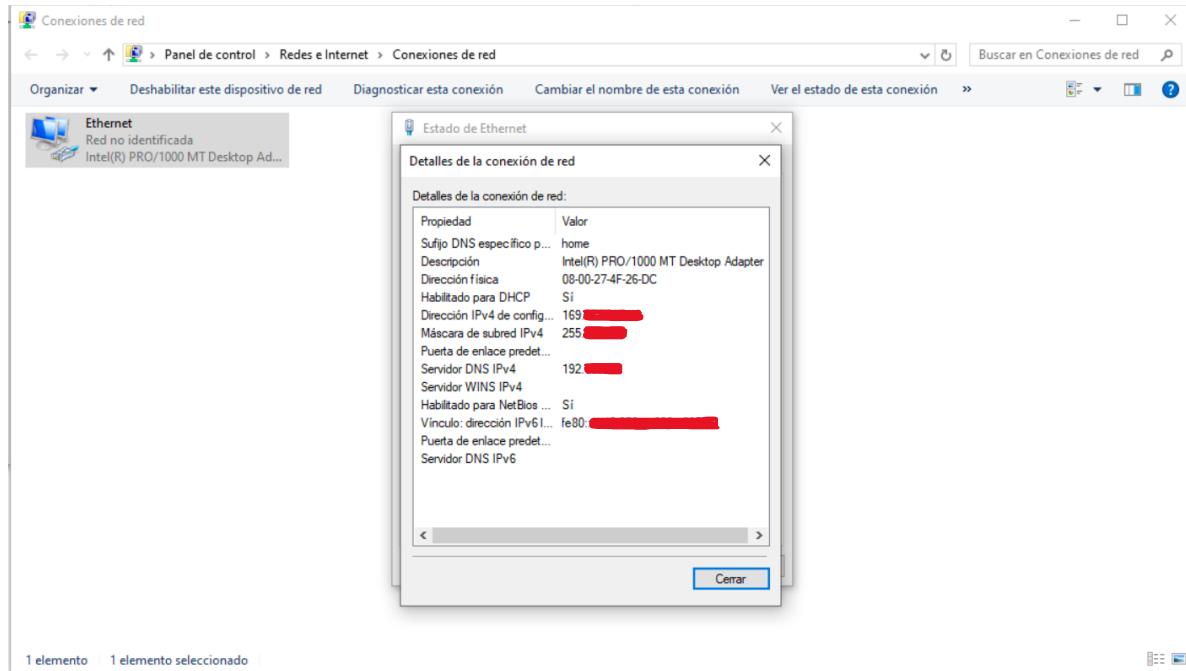
## Tarea 1:

Conocer la dirección IP de tu equipo (o de otros dispositivos) es muy importante para diversas tareas en el mundo de la ciberseguridad, como puede ser la identificación de dispositivos en la red, el rastreo de actividad sospechosa e incluso para el diagnóstico y resolución de problemas. En esta primera tarea hay que encontrar nuestra dirección IP, el nombre del host y el nombre de usuario de dos maneras diferentes, la primera de ellas utilizando la interfaz gráfica de Windows 10, y la segunda, a través de comandos en PowerShell / CMD.

- Interfaz gráfica:

Dirección IP: Para encontrar la dirección IP en Windows a través de la interfaz gráfica:

- Abre el “Panel de control” desde el menú de inicio o utilizando la función de búsqueda.
- Dentro del Panel de control, selecciona la categoría “Redes e Internet”.
- En la sección “Conexiones de red”, selecciona la conexión Ethernet activa haciendo clic en ella.
- Luego, en el menú de la parte superior, haz clic en “Detalles”.
- En la ventana de detalles de la conexión Ethernet, encontrarás información detallada sobre la configuración de red, incluida la dirección IP asignada a tu equipo.



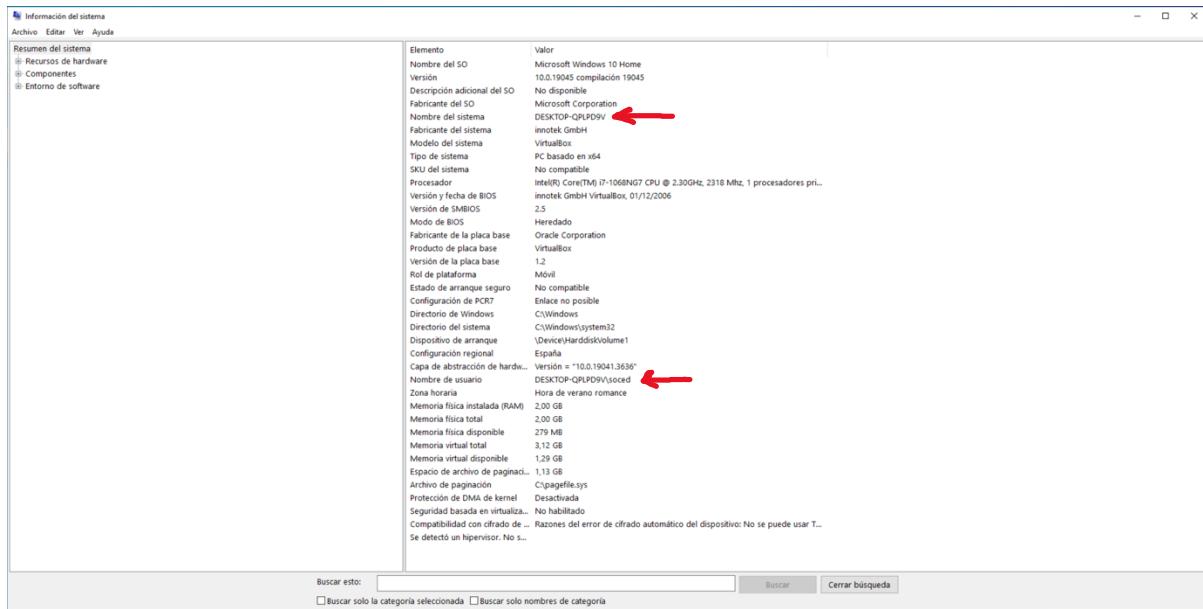
Nombre del host y Nombre de usuario: Para encontrar el nombre del host y el nombre de usuario en Windows, puedes seguir estos pasos:

- Haz clic en la barra de búsqueda en la barra de tareas.

- Escribe “Información del sistema” y presiona Enter.

Esto abrirá la ventana de “Información del sistema”, donde puedes encontrar la siguiente información:

- Nombre del sistema: Este campo mostrará el nombre asignado al equipo: “Desktop-QPLPD9V”.
- Nombre de usuario: En este campo se muestra el nombre de usuario con el que has iniciado sesión en el equipo, por ejemplo, “soced”.



- cmd:

Para acceder a la línea de comandos de Windows (cmd), escribe "cmd" en la barra de inicio y presiona Enter.

Una vez en la línea de comandos, puedes utilizar los siguientes comandos:

- Para conocer la dirección IP de tu equipo, escribe “ipconfig” y presiona Enter. Esto te mostrará la configuración de red de tu equipo, incluida su dirección IP.
- Para conocer el nombre del host y del usuario actual, escribe “whoami” y presiona Enter. Este comando te mostrará el nombre del host y el nombre de usuario con el que has iniciado sesión en tu equipo.

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.3803]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\soced>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . . : home
    Vínculo: dirección IPv6 local. . . : fe80::[REDACTED]
    Dirección IPv4 de configuración automática: 169.[REDACTED]
    Máscara de subred . . . . . : 255.[REDACTED]
    Puerta de enlace predeterminada . . . . . :

C:\Users\soced>whoami
desktop-qplpd9v\soced

C:\Users\soced>
```

## Tarea 2:

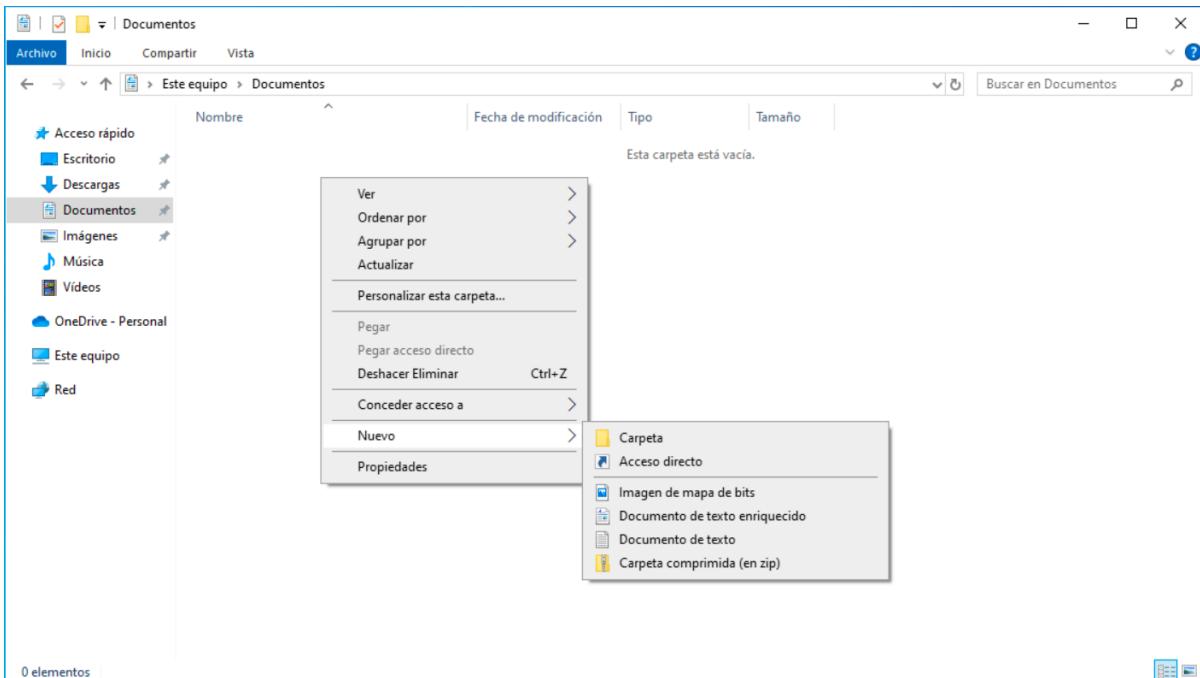
Una tarea muy importante y del día a día para un usuario de Windows (y por tanto para un analista de ciberseguridad también) es saber moverse bien por el explorador de archivos, crear archivos en alguna ruta específica, moverlo a otra diferente, copiarlo, pegarlo, borrarlo... Todo esto puede ayudar a un analista de seguridad para ser más rápido en la respuesta ante incidentes y conocer la estructura de los sistemas de archivos puede ayudar a comprender como los atacantes explotan vulnerabilidades. Pues bien, todo esto se puede hacer también a través de comandos de PowerShell y, con la soltura que se coge con la práctica, acaba siendo más fácil y práctico hacerlo de esta manera.

a) Por tanto, se pide en esta tarea crear una carpeta en la ruta “C:\Users\soced\Documents” llamada “Tarea\_2” utilizando la interfaz gráfica del explorador de archivos. Dentro de esta carpeta hay que crear un documento de texto llamado “tarea\_2”, también utilizando la GUI (Graphical User Interface). Lo abrimos y escribimos lo siguiente: “Texto de prueba para la tarea 2”.

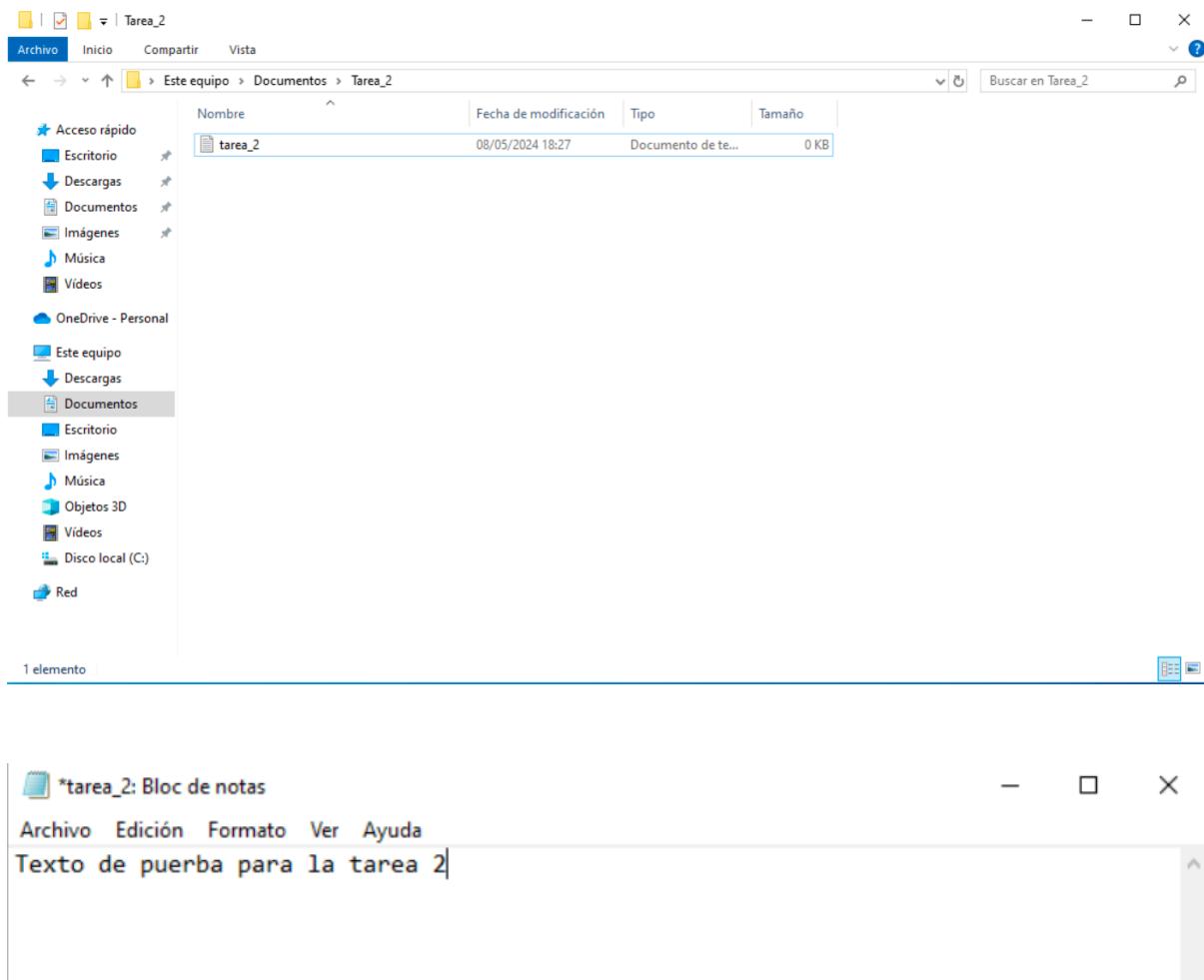
- Interfaz gráfica:

- En la barra de tareas, localiza y haz clic en el acceso directo al Explorador de archivos.
- Dentro del Explorador de archivos, navega hasta la ubicación de “Documentos”.
- Una vez dentro de la carpeta “Documentos”, haz clic con el botón derecho del ratón.
- En el menú que aparece, selecciona “Nuevo” y luego “Carpeta”.

- Se creará una nueva carpeta sin nombre. Asígnale el nombre “Tarea\_2” escribiéndolo y presionando Enter.



- Entra en la carpeta “Tarea\_2” que acabas de crear en la carpeta “Documentos”.
- Dentro de la carpeta “Tarea\_2”, haz clic con el botón derecho del ratón.
- En el menú que aparece, selecciona “Propiedades”.
- En la ventana de propiedades, haz clic en la pestaña “Nuevo”.
- Selecciona “Documento de texto” y asigna el nombre “tarea\_2” a este nuevo archivo.
- Haz doble clic con el botón izquierdo del ratón en el archivo “tarea\_2” para abrirlo y empezar a escribir el texto.



**b) Una vez realizadas estas tareas utilizando la interfaz gráfica, se pide realizar tareas similares, pero esta vez utilizando PowerShell. En primer lugar, hay que moverse hasta el Escritorio (Desktop), para listar el contenido que hay ahí. Una vez visto eso, se pide mover el archivo “Bienvenidos.txt” (si no existe crearlo a través de comandos de PowerShell) a la carpeta que se ha creado antes en “Documents” (Tarea\_2). Una vez se ha movido, mostrar el contenido de esa carpeta. Por último, hay que copiar el archivo de texto creado a través de la GUI (tarea\_2.txt) a uno nuevo llamado “tarea\_2\_cp.txt”, mostrar el texto que tiene dicho archivo de texto, y borrar el original.**

- PowerShell:
  - En la barra de inicio, escribe “PowerShell” para buscar la aplicación.
  - Una vez que aparezca la aplicación “Windows PowerShell” en los resultados de búsqueda, haz clic en ella para abrir la consola de PowerShell.
  - Dentro de la consola de PowerShell, utiliza el comando “cd” seguido de la ruta del directorio al que deseas moverte.

- Luego, utiliza el comando “Get-ChildItem” para mostrar el contenido del directorio actual. Esto te proporcionará una lista de los archivos y carpetas dentro del directorio en el que te encuentras.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/powershell

PS C:\Users\soced> cd .\Desktop
PS C:\Users\soced\Desktop> get-childitem

Directorio: C:\Users\soced\Desktop

Mode           LastWriteTime      Length Name
----           -----          ---- 
-a---  22/04/2024       8:54        190 Bienvenidos.txt
```

- A continuación, utiliza el comando “Move-Item” seguido de la ruta del archivo que deseas mover y la ruta de destino donde deseas moverlo.
- Para verificar que el archivo se haya movido correctamente, puedes utilizar el comando “cd” para navegar al directorio de destino y el comando “Get-ChildItem” para mostrar el contenido del directorio actual y asegurarte de que el archivo se haya movido correctamente.

```
Windows PowerShell
PS C:\Users\soced\Desktop> Move-Item -Path .\Bienvenidos.txt -Destination C:\Users\soced\Documents\Tarea_2
PS C:\Users\soced\Desktop> cd ..
PS C:\Users\soced> cd Documents
PS C:\Users\soced\Documents> cd Tarea_2
PS C:\Users\soced\Documents\Tarea_2> get-childitem

Directorio: C:\Users\soced\Documents\Tarea_2

Mode           LastWriteTime      Length Name
----           -----          ---- 
-a---  22/04/2024       8:54        190 Bienvenidos.txt
-a---  17/05/2024       10:46         31 tarea_2.txt
```

- Utiliza el comando “Copy-Item” seguido de la ruta del archivo que deseas copiar y la ruta de destino donde deseas copiarlo.
- “Get-Content” para mostrar el contenido del archivo copiado.
- “Remove-Item” para eliminar el archivo y “Get-ChildItem” para mostrar el contenido del directorio y comprobar así que el archivo se ha eliminado.

```
Seleccionar Windows PowerShell
PS C:\Users\soced\Documents\Tarea_2> copy-item -Path .\tarea_2.txt -Destination .\tarea_2_cp.txt
PS C:\Users\soced\Documents\Tarea_2> get-content -Path .\tarea_2_cp.txt
Texto de prueba para la tarea 2
PS C:\Users\soced\Documents\Tarea_2> remove-item -Path .\tarea_2.txt
PS C:\Users\soced\Documents\Tarea_2> get-childitem

Directorio: C:\Users\soced\Documents\Tarea_2

Mode           LastWriteTime      Length Name
----           -----          ---- 
-a---  22/04/2024     8:54        190 Bienvenidos.txt
-a---  17/05/2024    10:46         31 tarea_2_cp.txt

PS C:\Users\soced\Documents\Tarea_2>
```

## Tarea 3:

En este ejercicio se pretende continuar con los conceptos vistos en el ejercicio anterior, pero se introducen nuevos requisitos como pueden ser la búsqueda de contenido específico dentro de ellos. Para ello se pide crear un directorio llamado “Informes” en “C:\Temp” (si “C:\Temp” no existe, crearlo primero). Dentro del directorio “Informes” crear tres archivos de texto: Reporte1.txt, Reporte2.txt, y Reporte3.txt (con comandos de CMD o PowerShell). Tras esto escribir datos específicos en cada archivo (con comandos de CMD o PowerShell). Después hay que listar todos los archivos del directorio “Informes” y mostrar su tamaño en bytes, y por último, buscar y mostrar las líneas que contengan la palabra “error” en cualquiera de los archivos dentro de “Informes”.

- PowerShell:

Para empezar, verificamos si el directorio “C:\Temp” existe. Si no existe, lo creamos utilizando un condicional. Luego, creamos el directorio “Informes” dentro de “C:\Temp” utilizando el comando “New-Item”. Seguidamente, creamos tres archivos de texto llamados “Reporte1.txt”, “Reporte2.txt” y “Reporte3.txt” dentro del directorio “Informes”, utilizando nuevamente el comando “New-Item”.

Después de crear los archivos, añadimos contenido específico a cada uno de ellos utilizando el comando “Set-Content”.

```

Windows PowerShell

PS C:\Users\soced\Documents\Tarea_2> if (-Not (test-path -Path "C:\Temp")) {
>> new-item -Path "C:\Temp" -ItemType Directory
>> }

Directorio: C:\

Mode          LastWriteTime      Length Name
----          -----          ----  --
d---          17/05/2024       11:55          Temp

PS C:\Users\soced\Documents\Tarea_2> new-item -Path "C:\Temp\InFormes" -ItemType Directory

Directorio: C:\Temp

Mode          LastWriteTime      Length Name
----          -----          ----  --
d---          17/05/2024       11:57          InFormes

PS C:\Users\soced\Documents\Tarea_2> new-item -Path "C:\Temp\InFormes\Reporte1.txt" -ItemType File

Directorio: C:\Temp\InFormes

Mode          LastWriteTime      Length Name
----          -----          ----  --
-a--          17/05/2024       11:58          0 Reporte1.txt

PS C:\Users\soced\Documents\Tarea_2> new-item -Path "C:\Temp\InFormes\Reporte2.txt" -ItemType File

Directorio: C:\Temp\InFormes

Mode          LastWriteTime      Length Name
----          -----          ----  --
-a--          17/05/2024       11:59          0 Reporte2.txt

PS C:\Users\soced\Documents\Tarea_2> new-item -Path "C:\Temp\InFormes\Reporte3.txt" -ItemType File

Directorio: C:\Temp\InFormes

Mode          LastWriteTime      Length Name
----          -----          ----  --
-a--          17/05/2024       11:59          0 Reporte3.txt

PS C:\Users\soced\Documents\Tarea_2> set-content -Path "C:\Temp\InFormes\Reporte1.txt" -Value "Este es el reporte1."
PS C:\Users\soced\Documents\Tarea_2> set-content -Path "C:\Temp\InFormes\Reporte2.txt" -Value "Este es el reporte2."
PS C:\Users\soced\Documents\Tarea_2> set-content -Path "C:\Temp\InFormes\Reporte3.txt" -Value "Este es el reporte3."

```

Luego, nos movemos al directorio “Informes” utilizando el comando “Set-Location”. Una vez en este directorio, utilizamos el comando “Get-ChildItem” para listar todos los archivos presentes en él. Especificamos que nos interesa obtener el nombre del archivo y el tamaño en bytes de cada uno de ellos.

---

#### Windows PowerShell

```
PS C:\Users\soced\Documents\Tarea_2> set-location -Path "C:\Temp\Informes"
PS C:\Temp\Informes> get-childitem | select-object Name, Length

Name          Length
----          -----
Reporte1.txt    22
Reporte2.txt    22
Reporte3.txt    22

PS C:\Temp\Informes>
```

Como los textos deben contener la palabra error, añadimos más contenido a los archivos utilizando nuevamente el comando “Set-Content”.

---

#### Windows PowerShell

```
PS C:\Temp\Informes> add-content -Path .\Reporte1.txt -Value " No hay errores en este informe."
PS C:\Temp\Informes> add-content -Path .\Reporte2.txt -Value " Error encontrado."
PS C:\Temp\Informes> add-content -Path .\Reporte3.txt -Value " Nada mas que añadir."
PS C:\Temp\Informes> get-childitem | select-object Name, Length

Name          Length
----          -----
Reporte1.txt    56
Reporte2.txt    42
Reporte3.txt    45

PS C:\Temp\Informes>
```

Finalmente, utilizamos el comando “Get-ChildItem” con el filtro “\*.txt” para obtener una lista de todos los archivos con extensión “.txt” en el directorio actual y “ForEach-Object” para iterar sobre cada archivo “.txt”. En cada archivo, utilizamos el comando “Select-String” para buscar la palabra “error”. Utilizamos “\$\_.FullName” para proporcionar la ruta completa del archivo actual a “Select-String”, de modo que pueda buscar en el archivo correctamente. Esto nos permite mostrar todas las líneas que contienen la palabra “error” en cualquiera de los archivos dentro del directorio “Informes”.

```
➤ Seleccionar Windows PowerShell

PS C:\Temp\Informes> get-childitem -Filter *.txt | foreach-object {
>> select-string -Path $_.FullName -Pattern "error"
>> }

Reporte1.txt:2: No hay errores en este informe.
Reporte2.txt:2: Error encontrado.

PS C:\Temp\Informes>
```

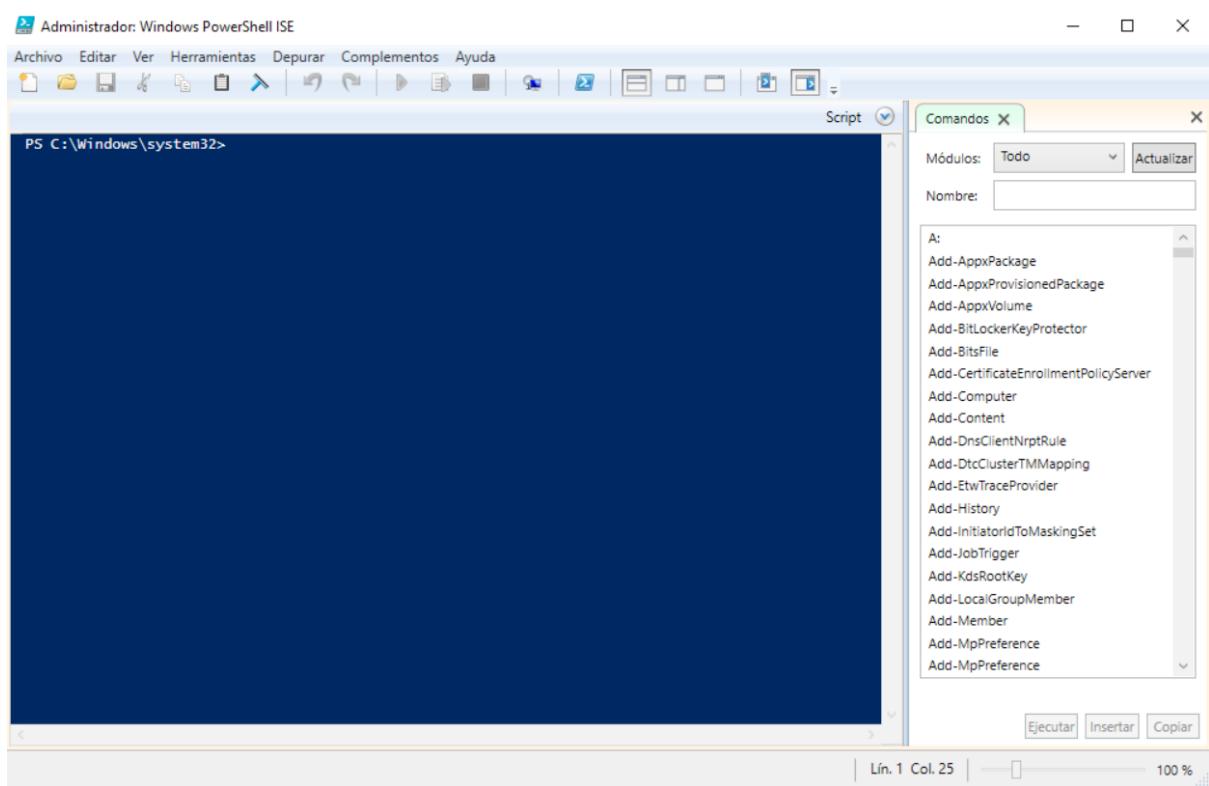
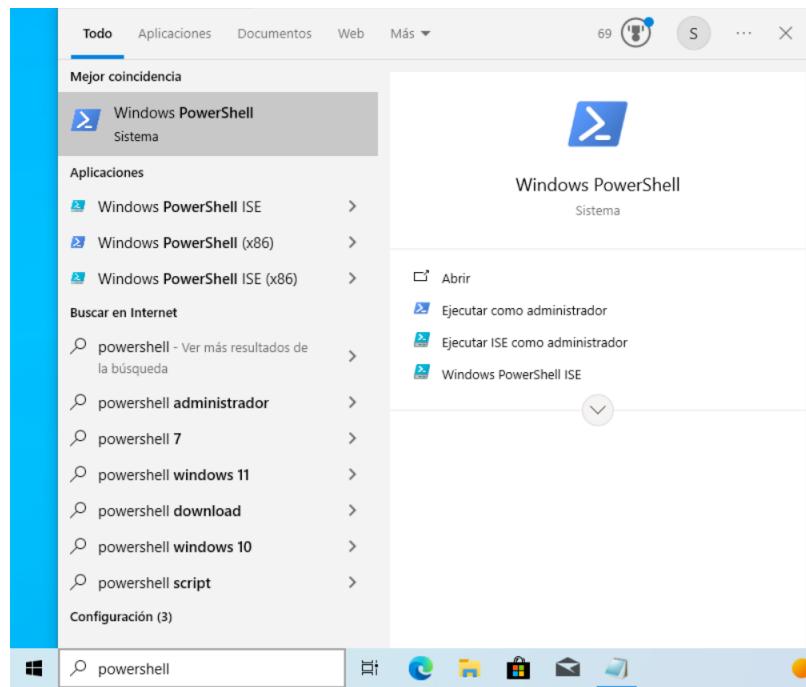
## Tarea 4:

En esta tarea se pide realizar un script de PowerShell que genere un documento de tipo texto (.txt) del sistema, en el que se incluya información sobre la CPU (información y velocidad), la memoria RAM (total y disponible), el espacio en el disco (tamaño y espacio disponible) y la configuración de red (nombre del adaptador, dirección IP y dirección MAC). Para poder ejecutar un script de PowerShell, como se ha visto en la teoría, es necesario abrir el PowerShell (o PowerShell ISE) como administrador y permitir la ejecución de scripts (REVISAR TEORÍA). Una vez realizado el script y obtenido el documento de texto con la información del sistema, se pide contrastar la información obtenida con la que se puede obtener desde el “Administrador de Tareas” de Windows.

- PowerShell:

Para acceder a PowerShell ISE con permisos de administrador, sigue estos pasos:

- Escribe “PowerShell” en el cuadro de búsqueda del menú de inicio.
- Selecciona “Ejecutar ISE como administrador”.
- Acepta los permisos necesarios para ejecutar PowerShell ISE con privilegios de administrador.



Un script de PowerShell puede ser escrito con el bloc de notas y guardado con la extensión “.ps1”.

Podemos usar PowerShell para obtener información del sistema a través de las Clases Win32.

Las clases Win32 son parte de la infraestructura de Windows Management Instrumentation (WMI). WMI es una implementación de la iniciativa Web-Based Enterprise Management (WBEM) y la especificación Common Information Model (CIM). Estas clases proporcionan una interfaz estándar para obtener información sobre la configuración, el estado y los componentes del sistema operativo y el hardware.

Dentro de estas, la clase Win32\_Processor representa los procesadores instalados en el sistema. Usamos el comando “get-ciminstance” para obtener instancias de las clases de WMI, en este caso Win32\_Processor seleccionando la información que nos interesa.

- CPU: nombre del procesador, y velocidad máxima del mismo en MHz.

```
$cpu = Get-CimInstance -ClassName Win32_Processor | Select-Object -Property Name, MaxClockSpeed
```

- RAM: Para la Memoria RAM, hemos seleccionado la cantidad de memoria visible en kilobytes y la cantidad de memoria libre en kilobytes.

```
$ram = Get-CimInstance -ClassName Win32_OperatingSystem | Select-Object -Property TotalVisibleMemorySize, FreePhysicalMemory
```

- Disco: En el caso del espacio en disco, buscamos en los discos lógicos del sistema, filtrando solo los discos de tipo 3 (discos duros locales) y seleccionamos el identificador del dispositivo, tamaño total del disco en bytes y espacio libre en el disco en bytes.

```
$disks = Get-CimInstance -ClassName Win32_LogicalDisk -Filter "DriveType=3" | Select-Object -Property DeviceID, Size, FreeSpace
```

- Red: Sobre la configuración de red, filtramos los adaptadores de red que estén habilitados para IP y seleccionamos la descripción del adaptador de red, la dirección MAC del adaptador y la dirección IP asociada al adaptador.

```
$networkAdapters = Get-CimInstance -ClassName Win32_NetworkAdapterConfiguration | Where-Object { $_.IPEnabled -eq $true } | Select-Object -Property Description, MACAddress, IPAddress
```

Posteriormente, añadimos toda esta información obtenida en un archivo que guardamos en el directorio “Temp” y mostramos su contenido por pantalla.

```

SystemInfo: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Obtener información sobre la CPU
$cpu = Get-CimInstance -ClassName Win32_Processor | Select-Object -Property Name, MaxClockSpeed

# Obtener información sobre la memoria RAM
$ram = Get-CimInstance -ClassName Win32_OperatingSystem | Select-Object -Property TotalVisibleMemorySize, FreePhysicalMemory

# Obtener información sobre el espacio en el disco
$disks = Get-CimInstance -ClassName Win32_LogicalDisk -Filter "DriveType=3" | Select-Object -Property DeviceID, Size, FreeSpace

# Obtener información sobre la configuración de red
$networkAdapters = Get-CimInstance -ClassName Win32_NetworkAdapterConfiguration | Where-Object { $_.IPEnabled -eq $true } | Select-Object -Property Description, MACAddress, IPAddress

# Crear un archivo de texto para almacenar la información
$outputFile = "C:\Temp\SystemInfo.txt"

# Escribir la información en el archivo
Add-Content -Path $outputFile -Value "Información de la CPU:"
foreach ($c in $cpu) {
    Add-Content -Path $outputFile -Value "Nombre: $($c.Name)"
    Add-Content -Path $outputFile -Value "Velocidad Máxima: $($c.MaxClockSpeed) MHz"
}

Add-Content -Path $outputFile -Value "`nInformación de la Memoria RAM:"
foreach ($r in $ram) {
    $totalMemoryGB = [math]::round($r.TotalVisibleMemorySize / 1MB, 2)
    $freeMemoryGB = [math]::round($r.FreePhysicalMemory / 1MB, 2)
    Add-Content -Path $outputFile -Value "Total de Memoria Visible: $totalMemoryGB GB"
    Add-Content -Path $outputFile -Value "Memoria Física Disponible: $freeMemoryGB GB"
}

Add-Content -Path $outputFile -Value "`nInformación del Espacio en el Disco:"
foreach ($d in $disks) {
    $totalDiskGB = [math]::round($d.Size / 1GB, 2)
    $freeDiskGB = [math]::round($d.FreeSpace / 1GB, 2)
    Add-Content -Path $outputFile -Value "Disco: $($d.DeviceID)"
    Add-Content -Path $outputFile -Value "Tamaño: $totalDiskGB GB"
    Add-Content -Path $outputFile -Value "Espacio Disponible: $freeDiskGB GB"
}

Add-Content -Path $outputFile -Value "`nConfiguración de Red:"
foreach ($n in $networkAdapters) {
    Add-Content -Path $outputFile -Value "Adaptador: $($n.Description)"
    Add-Content -Path $outputFile -Value "Dirección MAC: $($n.MACAddress)"
    Add-Content -Path $outputFile -Value "Dirección IP: $($n.IPAddress -join ', ')"
}

# Mostrar el contenido del archivo generado
Get-Content -Path $outputFile

```

Para ejecutarlo, primero nos movemos al directorio donde se encuentra el script con el comando “cd” y lo ejecutamos de este modo: “.\SystemInfo.ps1”. Al ejecutarlo, aparece un mensaje de advertencia debido a que el archivo no tiene permisos de ejecución. Al usar el comando “Set-ExecutionPolicy unrestricted”, aprendido en el curso, aparece un mensaje de seguridad que advierte sobre los peligros de aceptar y evitar la ejecución de archivos maliciosos. Quise investigar si había otro modo de ejecutarlo con menos riesgo y encontré “Bypass”, con el cual se le da acceso de ejecución puntual.

The screenshot shows the Windows PowerShell ISE interface. The main window displays a PowerShell session output:

```

PS C:\Windows\system32> cd C:\Users\soced\Desktop
PS C:\Users\soced\Desktop> .\SystemInfo.ps1
.\SystemInfo.ps1 : No se puede cargar el archivo C:\Users\soced\Desktop\SystemInfo.ps1 porque
la ejecución de scripts está deshabilitada en este sistema. Para obtener más información,
consulta el tema about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170.
En línea: 1 Carácter: 1
+ .\SystemInfo.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: () [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess

PS C:\Users\soced\Desktop> Set-ExecutionPolicy unrestricted
PS C:\Users\soced\Desktop> powershell -ExecutionPolicy Bypass -File .\SystemInfo.ps1
Información de la CPU:
Nombre: Intel(R) Core(TM) i7-1068NG7 CPU @ 2.30GHz
Velocidad MÁXIMA: 2298 MHz

Información de la Memoria RAM:
Total de Memoria Visible: 2 GB
Memoria FÍSICA Disponible: 0.59 GB

Información del Espacio en el Disco:
Disco: C:
Tamaño: 49.4 GB
Espacio Disponible: 27.24 GB

Configuración de Red:
Adaptador: Intel(R) PRO/1000 MT Desktop Adapter
Dirección MAC: 08:00:27:4F:26:DC
Dirección IP: 10.0.2.15, fe80::ea40:953e:c696:c893

```

To the right of the main window, there is a 'Comandos' (Commands) pane containing a list of cmdlets:

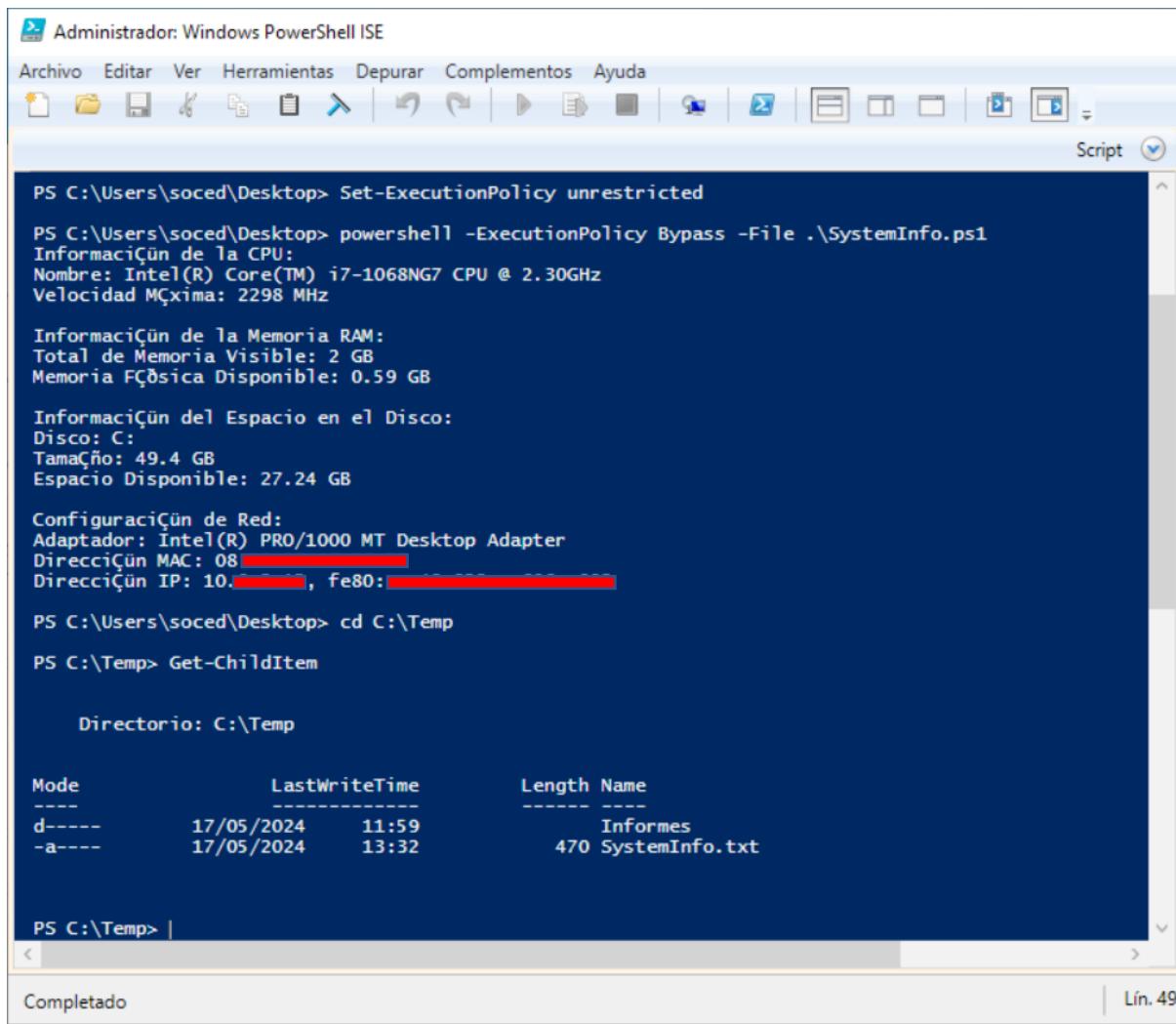
- A:
- Add-AppxPackage
- Add-AppxProvisionedPackage
- Add-AppxVolume
- Add-BitLockerKeyProtector
- Add-BitsFile
- Add-CertificateEnrollmentPolicyServer
- Add-Computer
- Add-Content
- Add-DnsClientNrrtRule
- Add-DtcClusterTMMapping
- Add-EtwTraceProvider
- Add-History
- Add-InitiatorIdToMaskingSet
- Add-JobTrigger
- Add-KdsRootKey
- Add-LocalGroupMember
- Add-Member
- Add-MpPreference
- Add-MpPreference

At the bottom of the interface, there are buttons for 'Ejecutar' (Execute), 'Insertar' (Insert), and 'Copiar' (Copy). The status bar shows 'Lín. 34 Col. 28' and '100 %'.

Hay varias políticas de ejecución disponibles:

- Restricted: No permite la ejecución de scripts.
- AllSigned: Permite la ejecución de scripts que están firmados por un editor de confianza.
- RemoteSigned: Permite la ejecución de scripts descargados de internet solo si están firmados por un editor de confianza.
- Unrestricted: Permite la ejecución de todos los scripts, pero muestra una advertencia antes de ejecutar scripts descargados de internet.
- Bypass: No bloquea nada y no muestra ninguna advertencia o solicitud de permiso. Útil para una ejecución específica.

Comprobamos la ubicación del archivo creado.



The screenshot shows a Windows PowerShell ISE window with the following content:

```
PS C:\Users\soced\Desktop> Set-ExecutionPolicy unrestricted
PS C:\Users\soced\Desktop> powershell -ExecutionPolicy Bypass -File .\SystemInfo.ps1
Informaci n de la CPU:
Nombre: Intel(R) Core(TM) i7-1068NG7 CPU @ 2.30GHz
Velocidad M xima: 2298 MHz

Informaci n de la Memoria RAM:
Total de Memoria Visible: 2 GB
Memoria F sica Disponible: 0.59 GB

Informaci n del Espacio en el Disco:
Disco: C:
Tama o: 49.4 GB
Espacio Disponible: 27.24 GB

Configuraci n de Red:
Adaptador: Intel(R) PRO/1000 MT Desktop Adapter
Direcci n MAC: 08:00:00:00:00:00
Direcci n IP: 10.0.0.1, fe80::0800:fffe%1

PS C:\Users\soced\Desktop> cd C:\Temp
PS C:\Temp> Get-ChildItem

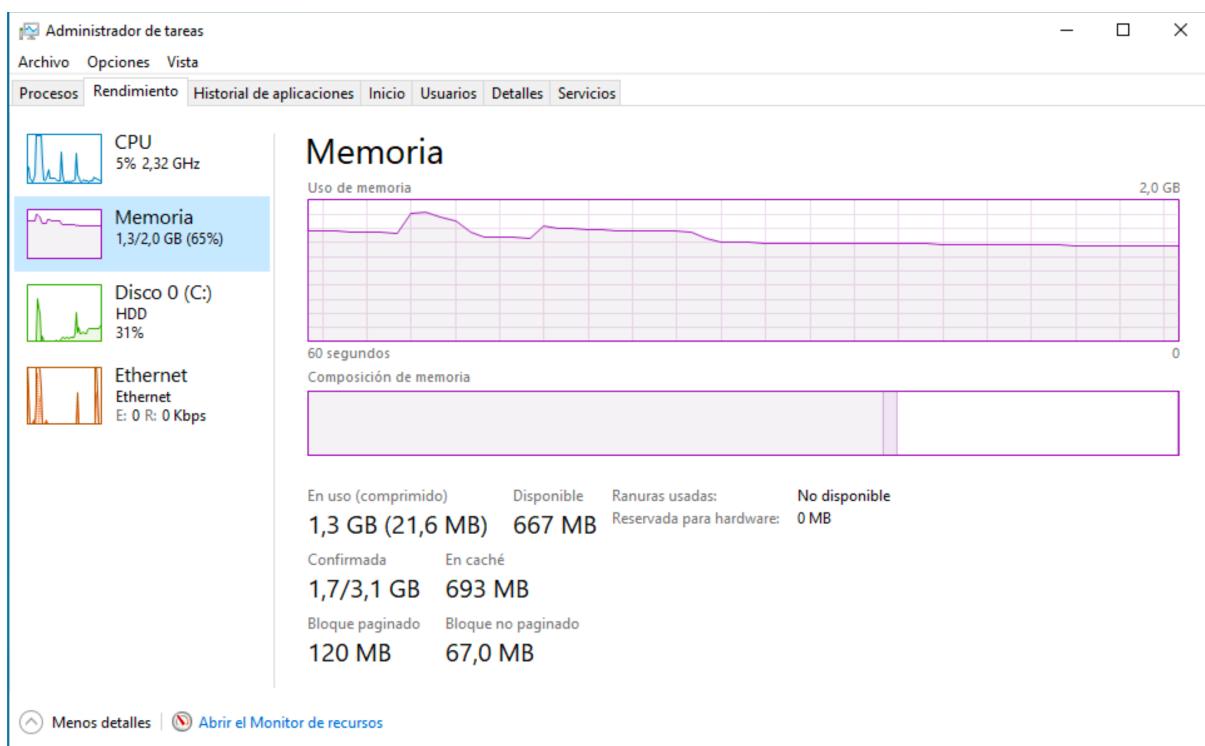
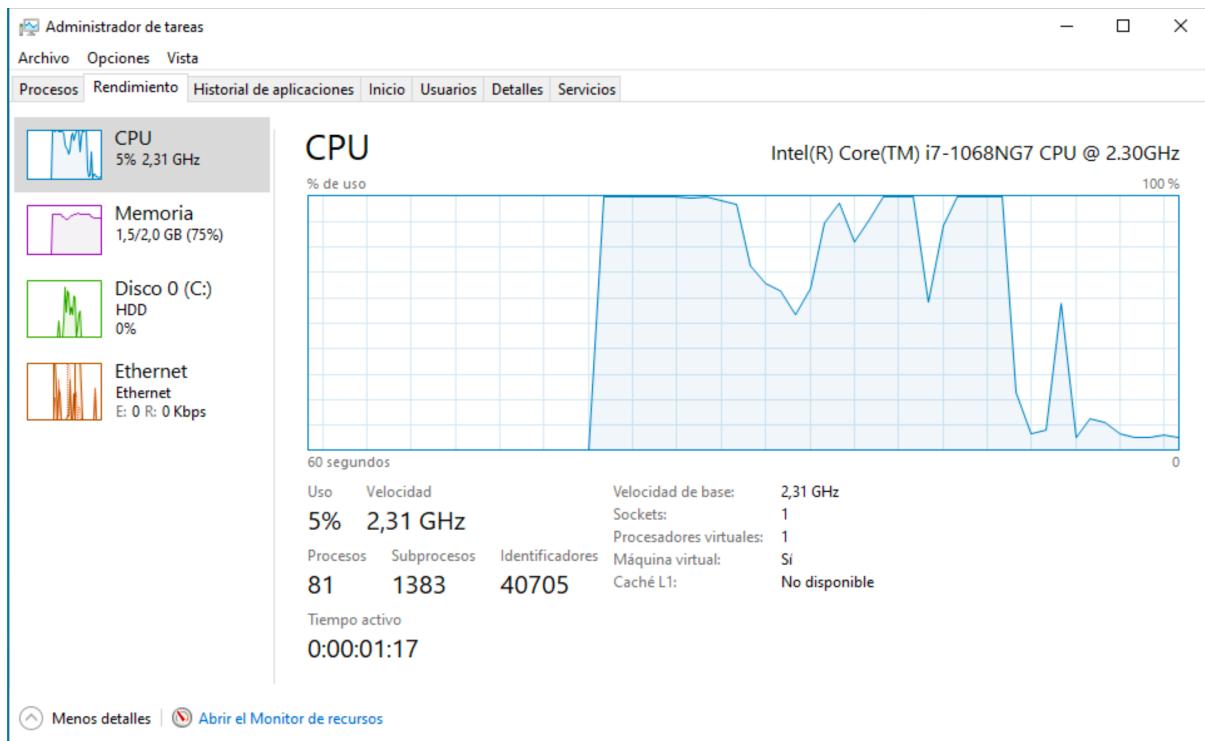
Directorio: C:\Temp

Mode LastWriteTime Length Name
---- ----- ----- ----
d--- 17/05/2024 11:59 In informes
-a-- 17/05/2024 13:32 470 SystemInfo.txt
```

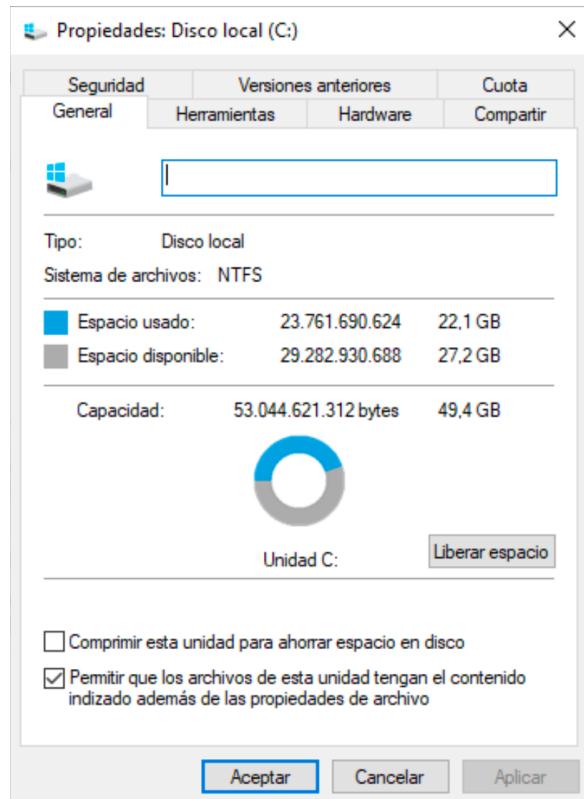
PS C:\Temp> |

Completado L n. 49

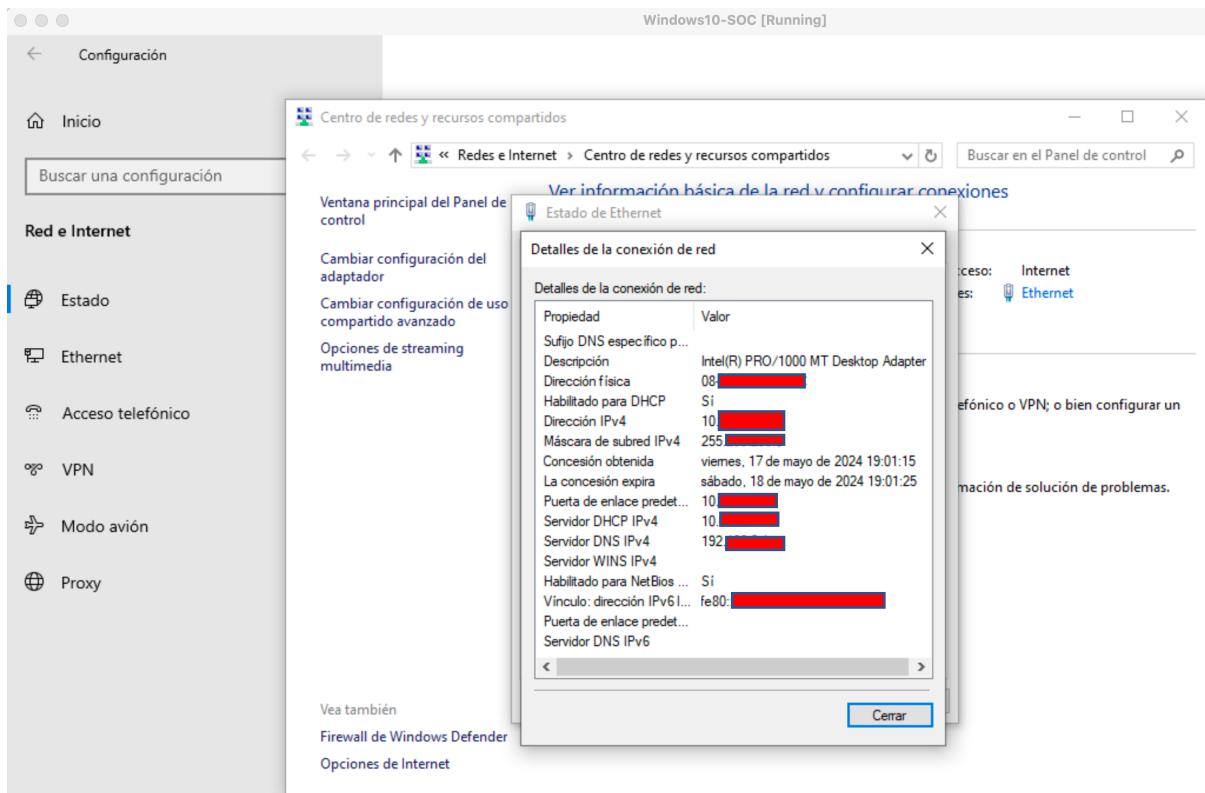
- Interfaz gr fica:
- CPU y RAM: Para comprobarlo desde la interfaz, nos dirigimos al administrador de tareas desde donde se puede acceder al rendimiento de la maquina, si seleccionamos la CPU podemos ver el nombre y velocidad del procesador y en la secci n de la memoria, la cantidad total de memoria y memoria disponible.



- **Disco:** Nos dirigimos al Explorador de archivos y seleccionamos “Este equipo”. Luego, hacemos clic derecho en el disco C:\ y seleccionamos “Propiedades” para ver el tamaño total del disco y el espacio disponible.



- **Red:** Nos dirigimos al menú de inicio y seleccionamos “Configuración”. Luego, hacemos clic en “Red e Internet” y seleccionamos “Centro de redes y recursos compartidos”. Finalmente, seleccionamos la conexión Ethernet y hacemos clic en "Detalles" para ver la dirección IP y la dirección MAC.



## Tarea 5:

**Otra tarea muy común en el ámbito de la ciberseguridad es ver los puertos que están en escucha y cuales no de un equipo. Los puertos de red de un dispositivo son identificadores numéricos que sirven para diferenciar las múltiples conexiones de red que pueden darse de manera simultánea en un dispositivo. Cada puerto está asociado con un número específico que permite a las aplicaciones de red enviar y recibir datos específicamente destinados a ellas. Esto es importante en el mundo de la ciberseguridad porque los puertos que están abiertos son un punto de entrada para atacantes, lo que hace que sea esencial conocer el estado de estos y realizar tareas de monitorización para detectar actividad sospechosa. En este ejercicio se desea crear un script de PowerShell básico capaz de leer los puertos actuales en escucha en el sistema (TCP). Para ello podremos usar el comando Get-NetTCPConnection -State Listen. Posteriormente, el objetivo será añadirlos a un documento de texto y guardararlo en una ruta concreta. Se recomienda hacer uso de Add-Content y un bucle foreach.**

Para cumplir con los requisitos de la tarea, podemos crear un script de PowerShell que utilice el comando “Get-NetTCPConnection” para obtener los puertos en escucha y luego guardar esta información en un archivo de texto en el directorio “Temp”.

```

ports: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Obtener conexiones TCP en estado Listen
$connections = Get-NetTCPConnection -State Listen

# Definir la ruta del archivo de salida
$outputFile = "C:\Temp\ListeningPorts.txt"

# Añadir encabezado al archivo de salida
Add-Content -Path $outputFile -Value "Puertos TCP en Escucha:"

# Añadir información de cada conexión al archivo
foreach ($conn in $connections) {
    $line = "Local Address: $($conn.LocalAddress):$($conn.LocalPort) - State: $($conn.State)"
    Add-Content -Path $outputFile -Value $line
}

# Verificar el contenido del archivo
Get-Content -Path $outputFile

```

Línea 15, columna 1 | 100% | UNIX (LF) | UTF-8

Observamos los puertos que están abiertos y por tanto receptivos a una conexión, algo importante para el monitoreo y análisis de ciberseguridad.

```

Administrador: Windows PowerShell ISE
Archivo Editar Ver Herramientas Depurar Complementos Ayuda
Script Script

PS C:\Windows\system32> cd C:\Users\soced\Desktop
PS C:\Users\soced\Desktop> powershell -executionpolicy Bypass -File .\ports.ps1
Puertos TCP en Escucha:
Local Address: :::49671 - State: Listen
Local Address: :::49669 - State: Listen
Local Address: :::49667 - State: Listen
Local Address: :::49666 - State: Listen
Local Address: :::49665 - State: Listen
Local Address: :::49664 - State: Listen
Local Address: :::5357 - State: Listen
Local Address: :::445 - State: Listen
Local Address: :::135 - State: Listen
Local Address: 0.0.0.0:49671 - State: Listen
Local Address: 0.0.0.0:49669 - State: Listen
Local Address: 0.0.0.0:49667 - State: Listen
Local Address: 0.0.0.0:49666 - State: Listen
Local Address: 0.0.0.0:49665 - State: Listen
Local Address: 0.0.0.0:49664 - State: Listen
Local Address: 0.0.0.0:5040 - State: Listen
Local Address: 10.0.2.15:139 - State: Listen
Local Address: 0.0.0.0:135 - State: Listen

PS C:\Users\soced\Desktop> cd C:\Temp
PS C:\Temp> Get-ChildItem

    Directorio: C:\Temp

Mode LastWriteTime Length Name
---- -- -- -- --
d---- 17/05/2024 11:59 Informes
-a--- 17/05/2024 15:05 800 ListeningPorts.txt
-a--- 17/05/2024 13:32 470 SystemInfo.txt

```

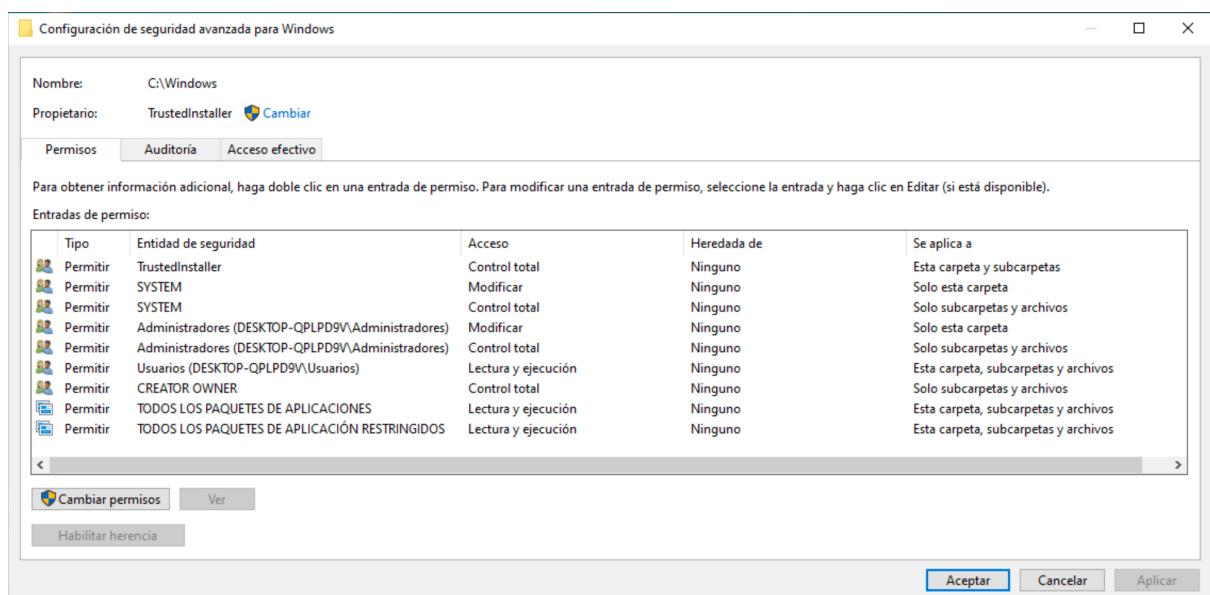
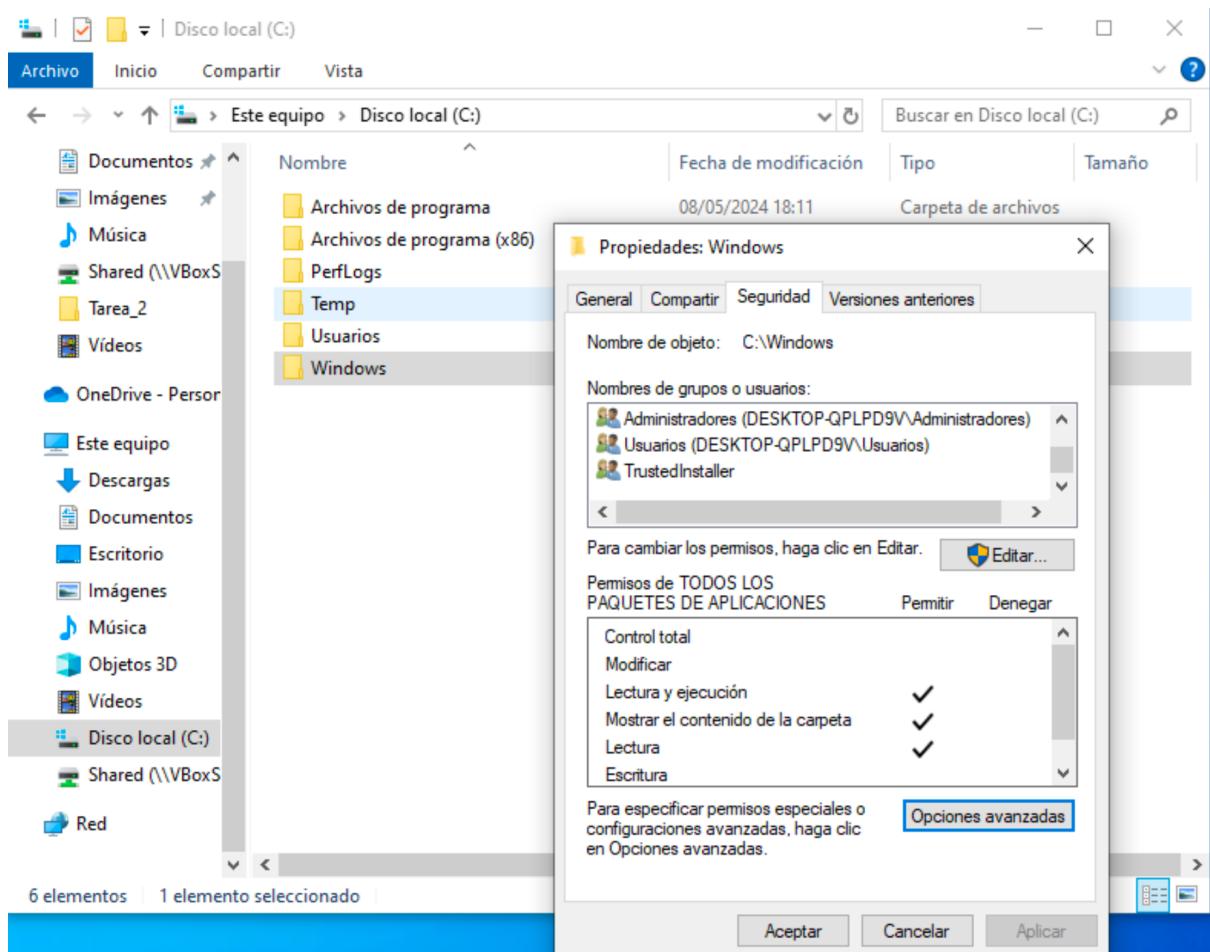
## Tarea 6:

**Los permisos en Windows son reglas asignadas a archivos y carpetas que determinan quién puede interactuar con ellos y de qué manera, permitiendo o no leer, modificar, escribir y/o eliminarlos. Esto hace que los permisos sean otro de esos conceptos básicos en la ciberseguridad, ya que una mala configuración en los permisos puede hacer que un atacante obtenga acceso a datos confidenciales o incluso a obtener acceso a todo el equipo, lo cual resultaría ser una gran amenaza. Estos permisos se pueden ver de diversas maneras, tanto utilizando la interfaz gráfica como comandos de PowerShell. Se pide comprobar con ambos métodos los permisos de la carpeta “Windows” (C:\Windows) para todos los grupos o usuarios y compararlos para ver si coinciden al hacerlo de estas dos maneras diferentes.**

- Interfaz gráfica:

En la interfaz gráfica del Explorador de archivos, podemos acceder a la lista de control de acceso (ACL) de una carpeta siguiendo estos pasos:

- Abrimos el Explorador de archivos y navegamos hasta "Este equipo".
- Seleccionamos el disco local C:\ y hacemos clic derecho.
- En el menú contextual, elegimos "Propiedades".
- En la ventana de propiedades, seleccionamos la pestaña "Seguridad".
- Hacemos clic en "Opciones avanzadas" para acceder a la configuración avanzada de seguridad.



- PowerShell:

En PowerShell, podemos obtener la misma información utilizando el siguiente comando:

```
Get-Acl -Path C:\Windows | Format-List
```

Este comando nos proporciona la lista de control de acceso (ACL) de la carpeta Windows, y al utilizar “Format-List”, formatea la salida para una mejor legibilidad.

Aunque la interfaz gráfica puede proporcionar una visualización más intuitiva de la información de seguridad, PowerShell nos brinda una forma rápida y eficiente de obtener la misma información.

Administrador: Windows PowerShell ISE

Archivo Editar Ver Herramientas Depurar Complementos Ayuda

PS C:\Windows\system32> Get-Acl -Path C:\Windows | Format-List

Path :	Microsoft.PowerShell.Core\FileSystem::C:\Windows
Owner :	NT SERVICE\TrustedInstaller
Group :	NT SERVICE\TrustedInstaller
Access :	CREATOR OWNER Allow 268435456 NT AUTHORITY\SYSTEM Allow 268435456 NT AUTHORITY\SYSTEM Allow Modify, Synchronize BUILTIN\Administradores Allow 268435456 BUILTIN\Administradores Allow Modify, Synchronize BUILTIN\Usuarios Allow -1610612736 BUILTIN\Usuarios Allow ReadAndExecute, Synchronize NT SERVICE\TrustedInstaller Allow 268435456 NT SERVICE\TrustedInstaller Allow FullControl ENTIDAD DE PAQUETES DE APLICACIONES\TODOS LOS PAQUETES DE APLICACIONES Allow ReadAndExecute, Synchronize ENTIDAD DE PAQUETES DE APLICACIONES\TODOS LOS PAQUETES DE APLICACIONES Allow -1610612736 ENTIDAD DE PAQUETES DE APLICACIONES\TODOS LOS PAQUETES DE APLICACION RESTRINGIDOS Allow ReadAndExecute, Synchronize ENTIDAD DE PAQUETES DE APLICACIONES\TODOS LOS PAQUETES DE APLICACION RESTRINGIDOS Allow -1610612736
Audit :	
Sddl :	O:S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464G:S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464D:PAI(A;OICIO;GA;;;CO)(A;OICIO;GA;;;SY)(A;0x1301bf;;;SY)(A;OICIO;GA;;;BA)(A; 0x1301bf;;;BA)(A;OICIO;GXGR;;;BU)(A;0x1200a9;;;BU)(A;CIO;GA;;;S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464)(A;;;FA;;;S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464)(A;0x1200a9;;;AC)(A;OICIO;GXGR;;;AC)(A;0x1200a9;;;S-1-15-2-2)(A;OICIO;GXGR;;;S-1-15-2-2)

PS C:\Windows\system32> |

## Tarea 7:

**Los hashes son algoritmos muy utilizados en el mundo de la ciberseguridad. En este ejercicio se pide investigar acerca de qué es un hash y qué utilidad tienen en la seguridad informática. Como extra, se recomienda describir los algoritmos de hashes que existen (los más comunes) y sus características y diferencias principales.**

## ¿Qué es un Hash?

Un hash es una función que toma una entrada (o mensaje) y devuelve una cadena de longitud fija, que normalmente parece ser una serie de caracteres alfanuméricos. Esta cadena se conoce como valor hash o hash. Los hashes son fundamentales en la seguridad informática por varias razones:

1. Integridad de los Datos: Los hashes se utilizan para verificar la integridad de los datos. Si dos conjuntos de datos tienen el mismo hash, es extremadamente probable que los datos sean idénticos.
2. Autenticación: Los hashes se usan para almacenar contraseñas de forma segura. En lugar de almacenar una contraseña en texto claro, se almacena el hash de la contraseña. Durante la autenticación, se compara el hash de la contraseña proporcionada con el hash almacenado.
3. Firmas Digitales: Los hashes se usan en firmas digitales para garantizar que un mensaje no ha sido alterado. El hash del mensaje se cifra con una clave privada para crear la firma digital.
4. Detección de Duplicados: Los hashes pueden identificar archivos duplicados en un sistema al comparar sus hashes.

## Algoritmos de Hash Comunes

Existen varios algoritmos de hash, cada uno con sus características y usos. Aquí describimos algunos de los más comunes:

### 1. MD5 (Message Digest Algorithm 5):

- Longitud del Hash: 128 bits (32 caracteres hexadecimales).
- Características: Fue muy popular debido a su rapidez, pero actualmente es considerado inseguro debido a vulnerabilidades que permiten colisiones (donde dos entradas diferentes generan el mismo hash).
- Uso: Se usaba para verificar la integridad de archivos y en contraseñas, pero ya no se recomienda para aplicaciones críticas de seguridad.

### 2. SHA-1 (Secure Hash Algorithm 1):

- Longitud del Hash: 160 bits (40 caracteres hexadecimales).
- Características: Mejor que MD5 en términos de seguridad, pero también se ha demostrado que tiene vulnerabilidades a colisiones.
- Uso: Usado en la autenticación y la integridad de datos, pero se está descontinuando en favor de algoritmos más seguros.

### 3. SHA-2 (Secure Hash Algorithm 2):

- Longitud del Hash: Variantes de 224, 256, 384, y 512 bits.
- Características: Es una familia de funciones hash que incluye SHA-224, SHA-256, SHA-384 y SHA-512. Mucho más seguro que MD5 y SHA-1.
- Uso: Amplia adopción en aplicaciones de seguridad modernas, incluyendo certificados SSL/TLS, firmas digitales, y en sistemas operativos.

#### **4. SHA-3 (Secure Hash Algorithm 3):**

- Longitud del Hash: Variantes de 224, 256, 384, y 512 bits.
- Características: Fue diseñado como un reemplazo a largo plazo para SHA-2, utilizando una estructura diferente conocida como "esponja".
- Uso: Todavía en proceso de adopción, se espera que se utilice ampliamente en el futuro.

#### **5. BLAKE2:**

- Longitud del Hash: Configurable, comúnmente 256 o 512 bits.
- Características: Diseñado para ser más rápido que MD5, SHA-1 y SHA-2, y tan seguro como SHA-3.
- Uso: Útil en aplicaciones donde la velocidad es crítica, como en hash tables y en la verificación de integridad de archivos.

#### **6. RIPEMD-160:**

- Longitud del Hash: 160 bits (40 caracteres hexadecimales).
- Características: Menos conocido, pero una alternativa segura a SHA-1.
- Uso: Utilizado en algunas aplicaciones de criptografía y firmas digitales.

#### **Comparación de Algoritmos de Hash**

Algoritmo	Longitud del Hash	Seguridad	Velocidad	Uso Actual
MD5	128 bits	Inseguro	Muy rápido	Evitar su uso en aplicaciones
SHA-1	160 bits	Inseguro	Rápido	Desaconsejado en favor de SHA-2 y SHA-3

SHA-2	224, 256, 384, 512 bits	Seguro	Moderado	Amplia adopción en seguridad
SHA-3	224, 256, 384, 512 bits	Muy seguro	Lento	Futuro reemplazo para SHA-2
BLAKE2	256, 512 bits	Muy seguro	Muy rápido	Aplicaciones de alta velocidad
RIPEMD-160	160 bits	Seguro	Moderado	Uso limitado en criptografía

## Tarea 8:

En esta tarea se pide obtener por PowerShell el hash de alguno de los documentos de texto generados en las tareas anteriores (también se puede realizar con documentos personales). Se recomienda hacer uso del comando Get-FileHash “documento.txt”. De esta manera, comprueba las siguientes cosas:

- a) Tamaño y algoritmo del hash que se obtiene por defecto de PowerShell.
- b) Comprobar si, para la misma entrada, se obtiene siempre el mismo hash.
- c) Modificar un solo carácter del documento de texto utilizado para generar los hashes para comprobar si el que se obtiene tras dicha modificación sigue siendo el mismo.
- d) Comprobar si, para distintos archivos se obtiene el mismo hash.

- PowerShell:

Para realizar esta tarea abrimos PowerShell como administrador. Elegimos el archivo “SystemInfo.txt” que guardamos anteriormente en el directorio C:\Temp.

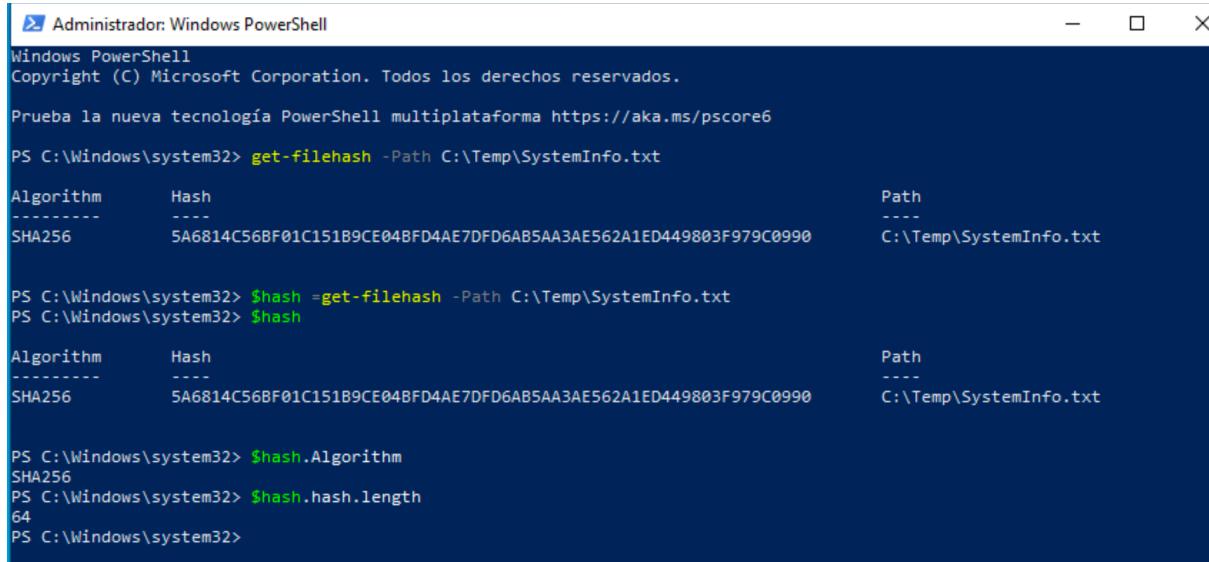
**a) Tamaño y algoritmo del hash que se obtiene por defecto de PowerShell.**

El comando "Get-FileHash" de PowerShell utiliza por defecto el algoritmo de hash SHA-256 para calcular el hash de un archivo. Para obtener información sobre el tamaño y el algoritmo del hash resultante, podemos guardar el hash obtenido por el comando en una variable, por ejemplo, \$hash, y luego utilizar los siguientes métodos para obtener los detalles:

Utilizando “\$hashAlgorithm”, podemos obtener el tipo de algoritmo utilizado, que en este caso es SHA256.

Mediante “\$hash.Hash.Length”, podemos determinar el tamaño del hash generado, que consiste en 64 caracteres hexadecimales, lo que equivale a 256 bits.

Esta información es valiosa para comprender el tipo de hash generado por defecto en PowerShell y su tamaño, lo que nos permite utilizarlo de manera efectiva en diversas aplicaciones de seguridad y verificación de integridad de archivos.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

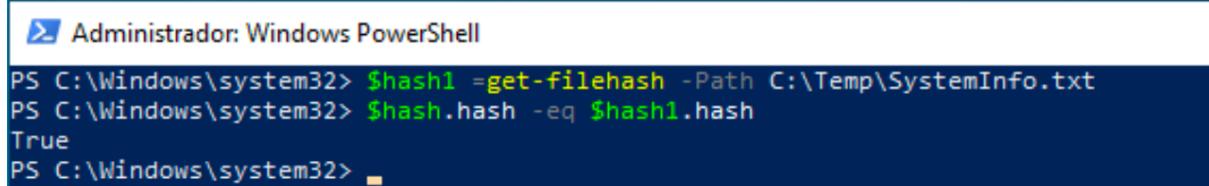
PS C:\Windows\system32> get-filehash -Path C:\Temp\SystemInfo.txt
Algorithm      Hash                               Path
-----        -----
SHA256         5A6814C56BF01C151B9CE04BFD4AE7DFD6AB5AA3AE562A1ED449803F979C0990
C:\Temp\SystemInfo.txt

PS C:\Windows\system32> $hash =get-filehash -Path C:\Temp\SystemInfo.txt
PS C:\Windows\system32> $hash
Algorithm      Hash                               Path
-----        -----
SHA256         5A6814C56BF01C151B9CE04BFD4AE7DFD6AB5AA3AE562A1ED449803F979C0990
C:\Temp\SystemInfo.txt

PS C:\Windows\system32> $hash.Algorithm
SHA256
PS C:\Windows\system32> $hash.hash.length
64
PS C:\Windows\system32>
```

### b) Comprobar si, para la misma entrada, se obtiene siempre el mismo hash.

En la captura previa, se observó que el hash fue calculado dos veces y que se mantuvo constante en ambas ocasiones. Para verificar esto, podemos emplear el siguiente comando:

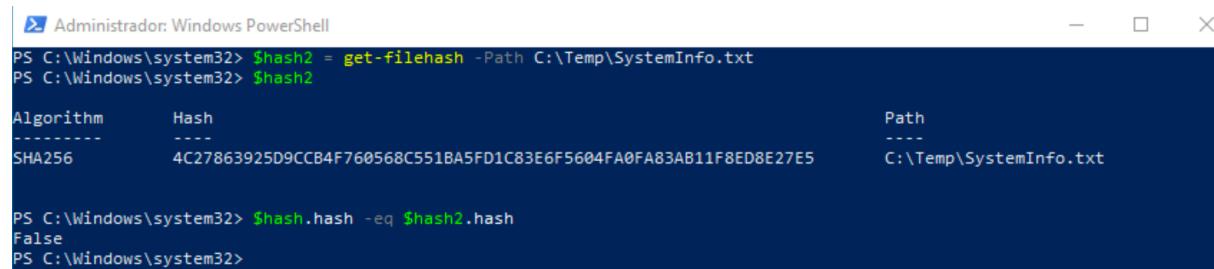


```
Administrator: Windows PowerShell
PS C:\Windows\system32> $hash1 =get-filehash -Path C:\Temp\SystemInfo.txt
PS C:\Windows\system32> $hash.hash -eq $hash1.hash
True
PS C:\Windows\system32>
```

Utilizamos el operador “-eq” en PowerShell, el cual significa “equal” (igual), para comparar si los dos hashes obtenidos son idénticos. Si el resultado de esta comparación es “True”, indica que ambos hashes son consistentes y no han cambiado. Esta verificación nos permite asegurarnos de que el hash generado para una misma entrada permanece inalterado, lo que es fundamental para garantizar la integridad y la autenticidad de los datos.

### c) Modificar un solo carácter del documento de texto utilizado para generar los hashes para comprobar si el que se obtiene tras dicha modificación sigue siendo el mismo.

Para llevar a cabo esta comprobación, primero modificamos manualmente un solo carácter en el documento de texto utilizando el Bloc de notas. Luego, procedemos a volver a calcular el hash del archivo modificado utilizando el mismo comando “Get-FileHash” en PowerShell.



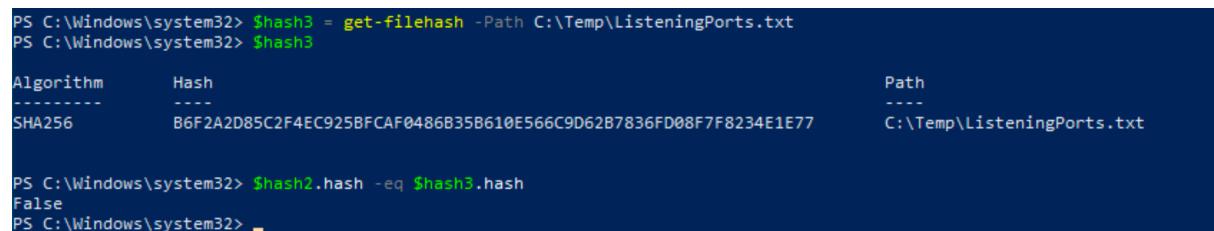
```
PS C:\Windows\system32> $hash2 = get-filehash -Path C:\Temp\SystemInfo.txt
PS C:\Windows\system32> $hash2
Algorithm      Hash                               Path
-----        -----
SHA256         4C27863925D9CCB4F760568C551BA5FD1C83E6F5604FA0FA83AB11F8ED8E27E5   C:\Temp\SystemInfo.txt

PS C:\Windows\system32> $hash.hash -eq $hash2.hash
False
PS C:\Windows\system32>
```

Al realizar esta operación, observamos que el hash resultante del archivo modificado será diferente al hash original. Este cambio en el hash indica que incluso una pequeña alteración en el contenido del archivo produce un efecto significativo en el resultado del hash. Esta propiedad es fundamental en el uso de funciones hash para la verificación de integridad de archivos, ya que nos permite detectar cualquier cambio, por mínimo que sea, en los datos originales.

#### d) Comprobar si, para distintos archivos se obtiene el mismo hash.

Para llevar a cabo esta comprobación, procedemos a obtener el hash del archivo “ListeningPorts.txt”, ubicado en el mismo directorio “Temp”, utilizando el mismo comando “Get-FileHash” en PowerShell. Luego, comparamos este hash con el hash obtenido anteriormente para el otro archivo.



```
PS C:\Windows\system32> $hash3 = get-filehash -Path C:\Temp\ListeningPorts.txt
PS C:\Windows\system32> $hash3
Algorithm      Hash                               Path
-----        -----
SHA256         B6F2A2D85C2F4EC925BFCAF0486B35B610E566C9D62B7836FD08F7F8234E1E77   C:\Temp\ListeningPorts.txt

PS C:\Windows\system32> $hash2.hash -eq $hash3.hash
False
PS C:\Windows\system32>
```

Al realizar esta comparación, podemos confirmar que los hashes de los diferentes archivos son distintos. Esta propiedad es fundamental para la identificación única de archivos.

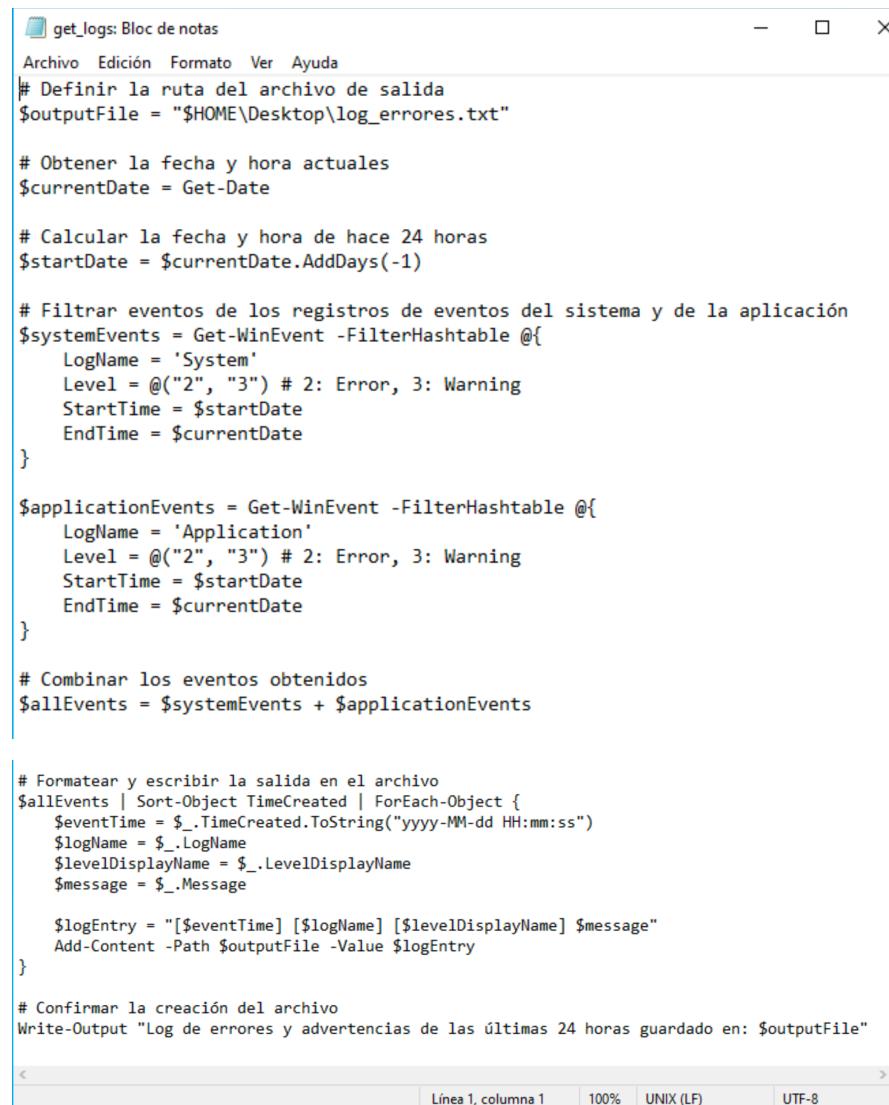
### Tarea 9:

**Los logs, o registros de eventos, son archivos que documentan actividades específicas dentro de un sistema o aplicación. Estos registros son fundamentales para el mantenimiento, el monitoreo y la seguridad de los sistemas informáticos, en el entorno de seguridad estos contienen registros relacionados con la seguridad del sistema operativo, como los inicios de sesión (tanto exitosos como fallidos), la gestión de cuentas de usuarios,**

**y políticas de seguridad. Crea un script de PowerShell que acceda a los logs de eventos de Windows y extraiga todos los errores y advertencias de las últimas 24 horas. Deberá formatear la salida en un documento de texto llamado “log\_errores.txt” guardado en el escritorio. Este tipo de script es útil para diagnósticos rápidos y respuesta a incidentes en ciberseguridad.**

- PowerShell:

Para comenzar, abrimos PowerShell en modo administrador. Luego, creamos un nuevo script llamado “get\_logs.ps1” utilizando el Bloc denotas. Una vez creado el script, nos dirigimos al escritorio utilizando el comando “set-location C:\Users\soced\Desktop” para asegurarnos de que estemos en la ubicación correcta. Finalmente, ejecutamos el script recién creado utilizando el comando adecuado en PowerShell. Este proceso nos permite ejecutar el script y obtener los resultados esperados.



```
get_logs: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Definir la ruta del archivo de salida
$outputFile = "$HOME\Desktop\log_errores.txt"

# Obtener la fecha y hora actuales
$currentTime = Get-Date

# Calcular la fecha y hora de hace 24 horas
$startDate = $currentTime.AddDays(-1)

# Filtrar eventos de los registros de eventos del sistema y de la aplicación
$systemEvents = Get-WinEvent -FilterHashtable @{
    LogName = 'System'
    Level = @("2", "3") # 2: Error, 3: Warning
    StartTime = $startDate
    EndTime = $currentTime
}

$applicationEvents = Get-WinEvent -FilterHashtable @{
    LogName = 'Application'
    Level = @("2", "3") # 2: Error, 3: Warning
    StartTime = $startDate
    EndTime = $currentTime
}

# Combinar los eventos obtenidos
$allEvents = $systemEvents + $applicationEvents

# Formatear y escribir la salida en el archivo
$allEvents | Sort-Object TimeCreated | ForEach-Object {
    $eventTime = $_.TimeCreated.ToString("yyyy-MM-dd HH:mm:ss")
    $logName = $_.LogName
    $levelDisplayName = $_.LevelDisplayName
    $message = $_.Message

    $logEntry = "[{$eventTime}] [{$_}] [{$_}] {$message}"
    Add-Content -Path $outputFile -Value $logEntry
}

# Confirmar la creación del archivo
Write-Output "Log de errores y advertencias de las últimas 24 horas guardado en: $outputFile"
```

Los comandos a destacar en este script son los usados para filtrar los eventos de los registros.

- En el caso de los eventos del sistema:

- “Get-WinEvent” obtiene los eventos de los registros de eventos de Windows.
- “-FilterHashtable”: utiliza un hashtable para especificar los criterios de filtro.
- “LogName = ‘System’”: Filtra los eventos del registro del sistema.
- “Level = @(“2”, “3”)": Filtra los eventos de nivel 2 (Error) y nivel 3 (Warning).
- “StartTime = \$startDate”: Especifica el tiempo de inicio del filtro (hace 24 horas).
- “EndTime = \$currentDate”: Especifica el tiempo de finalización del filtro (actual).

- Eventos de la Aplicación:

“LogName = ‘Application’”: Filtra los eventos del registro de la aplicación.

Los otros parámetros (Level, StartTime, EndTime) funcionan igual que en el filtro de eventos del sistema.

- Combinación de ambos:

“\$systemEvents + \$applicationEvents”: Combina los dos conjuntos de eventos obtenidos en una sola colección.

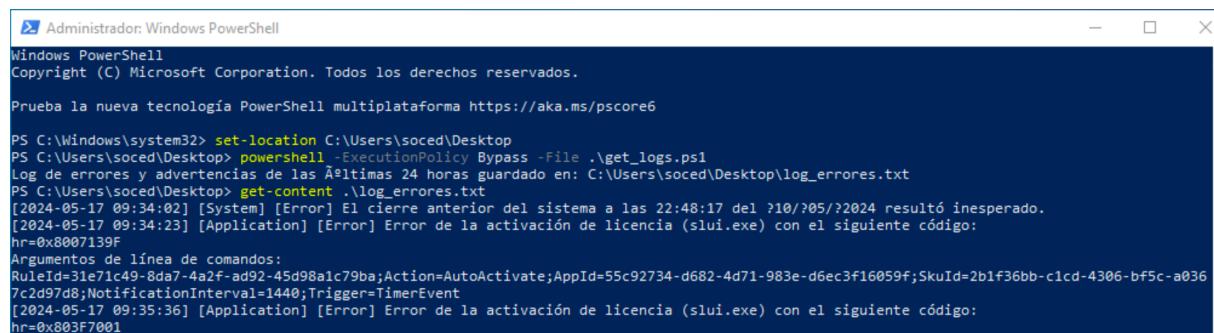
- Para formatear y escribir la salida en un archivo:

- “\$allEvents | Sort-Object TimeCreated”: Ordena los eventos por la propiedad “TimeCreated” (hora de creación del evento).
- “ForEach-Object { ... }”: Itera sobre cada evento en la colección ordenada.
- “\$\_ .TimeCreated.ToString(“yyyy-MM-dd HH:mm:ss”)": Formatea la fecha y hora de creación del evento en una cadena legible.
- “\$\_ .LogName”: Obtiene el nombre del registro del evento (por ejemplo, System o Application).
- “\$\_ .LevelDisplayName”: Obtiene el nombre descriptivo del nivel del evento (por ejemplo, Error o Warning).
- “\$\_ .Message”: Obtiene el mensaje del evento.
- “\$logEntry = “[ \$eventTime ] [ \$logName ] [ \$levelDisplayName ] \$message””: Construye una cadena de texto con la información del evento.

- “Add-Content -Path \$outputFile -Value \$logEntry”: Añade la cadena de texto construida al archivo de salida especificado.
- Por último, para confirmar la creación del archivo:

“Write-Output” imprime un mensaje en la consola de PowerShell, confirmando que el archivo de log ha sido guardado.

No obstante, he mostrado usando “get-content” el inicio del contenido del archivo obtenido; “log\_errores.txt”.



```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Windows\system32> set-location C:\Users\soced\Desktop
PS C:\Users\soced\Desktop> powershell -ExecutionPolicy Bypass -File .\get_logs.ps1
Log de errores y advertencias de las Últimas 24 horas guardado en: C:\Users\soced\Desktop\log_errores.txt
PS C:\Users\soced\Desktop> get-content .\log_errores.txt
[2024-05-17 09:34:02] [System] [Error] El cierre anterior del sistema a las 22:48:17 del 10/05/2024 resultó inesperado.
[2024-05-17 09:34:23] [Application] [Error] Error de la activación de licencia (slui.exe) con el siguiente código:
hr=0x8007139F
Argumentos de línea de comandos:
RuleId=31e71c49-8d7-4a2f-ad92-45d98a1c79ba;Action=AutoActivate;AppId=55c92734-d682-4d71-983e-d6ec3f16059f;SkuId=2b1f36bb-c1cd-4306-bf5c-a036
7c2d97d8;NotificationInterval=1440;Trigger=TimerEvent
[2024-05-17 09:35:36] [Application] [Error] Error de la activación de licencia (slui.exe) con el siguiente código:
hr=0x803F7001

```

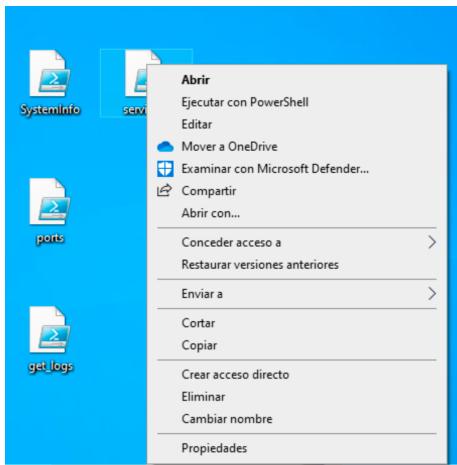
## Tarea 10:

Los servicios son aplicaciones en segundo plano que generalmente inician automáticamente al arrancar el sistema operativo y continúan ejecutándose en el fondo, realizando tareas esenciales para el funcionamiento del sistema o facilitando ciertas funcionalidades para otras aplicaciones. Se pide escribir un script de PowerShell que muestre todos los servicios de Windows que están configurados para iniciar automáticamente pero que actualmente están detenidos. El script debería ofrecer la opción de iniciar estos servicios a través de comandos adicionales. Esto es esencial para la administración de sistemas y aseguramiento de que servicios críticos estén en funcionamiento.

- PowerShell:

Para crear el script, nos dirigimos al Bloc de notas. Una vez allí, escribimos el código del script que deseamos crear. Una vez completado, guardamos el archivo con la extensión “.ps1” para indicar que es un script de PowerShell.

Para ejecutar el script, podemos hacerlo de varias maneras. Una opción es hacer clic derecho en el archivo del script y seleccionar “Ejecutar con PowerShell” en el menú.



○ PowerShell Script:

- El comando “Get-Service” obtiene todos los servicios del sistema, junto con la propiedad “StartType” se usa para filtrar los servicios configurados para iniciar automáticamente y “Status” para verificar si están detenidos.
- “Where-Object”: Filtra los servicios para aquellos que tienen “StartType” igual a “Automatic” y “Status” igual a “Stopped”.
- “If (\$stoppedAutoServices)": Verifica si hay servicios que coinciden con el filtro.
- “Write-Output”: Muestra un mensaje y la lista de servicios.
- “Read-Host”: Pregunta al usuario si desea iniciar los servicios.
- “If (\$startServices -eq ‘S’)": Verifica la respuesta del usuario.
- “Start-Service”: para iniciar los servicios seleccionados, en este caso, inicia cada servicio detenido.
- “Try/Catch”: Maneja errores potenciales al intentar iniciar un servicio.

```

servicios: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Obtener todos los servicios configurados para iniciar automáticamente pero que están detenidos
$stoppedAutoServices = Get-Service | Where-Object { $_.StartType -eq 'Automatic' -and $_.Status -eq 'Stopped' }

# Mostrar los servicios que estan detenidos
if ($stoppedAutoServices) {
    Write-Output "Los siguientes servicios estan configurados para iniciar automaticamente pero estan detenidos:"
    $stoppedAutoServices | ForEach-Object {
        Write-Output "$($_.DisplayName) ($($_.Name))"
    }
}

# Preguntar al usuario si desea iniciar los servicios
$startServices = Read-Host "¿Desea iniciar estos servicios? (S/N)"
if ($startServices -eq 'S') {
    # Iniciar los servicios
    $stoppedAutoServices | ForEach-Object {
        try {
            Start-Service -Name $_.Name -ErrorAction Stop
            Write-Output "Servicio '$($_.DisplayName)' iniciado."
        } catch {
            Write-Output "No se pudo iniciar el servicio '$($_.DisplayName)'. Error: $_"
        }
    }
} else {
    Write-Output "No se iniciaron los servicios."
}
} else {
    Write-Output "No hay servicios configurados para iniciar automaticamente que esten detenidos."
}

```

```

Windows PowerShell
Los siguientes servicios estan configurados para iniciar automaticamente pero estan detenidos:
Microsoft Edge Update Service (edgeupdate) (edgeupdate)
Administrador de mapas descargados (MapsBroker)
Protección de software (sppsvc)
¿Desea iniciar estos servicios? (S/N):

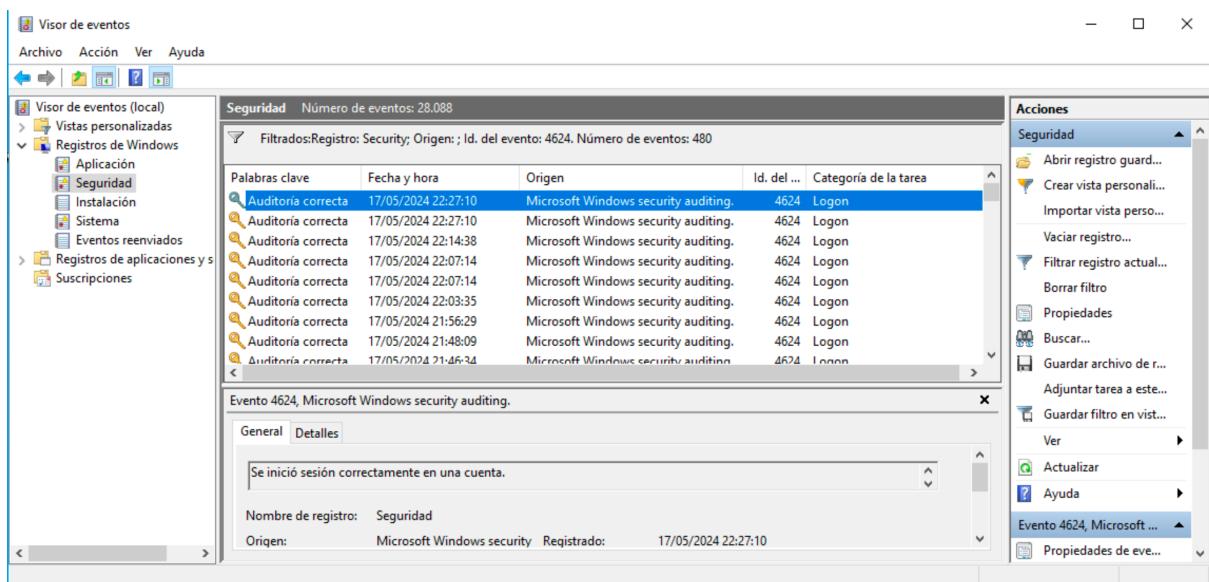
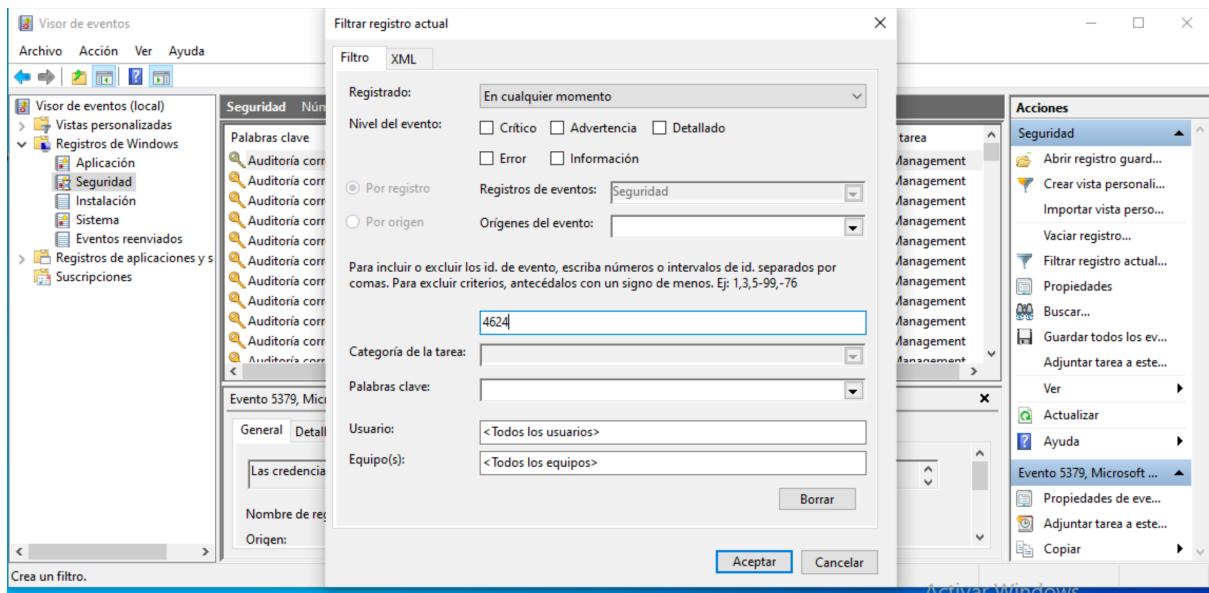
```

## Tarea 11:

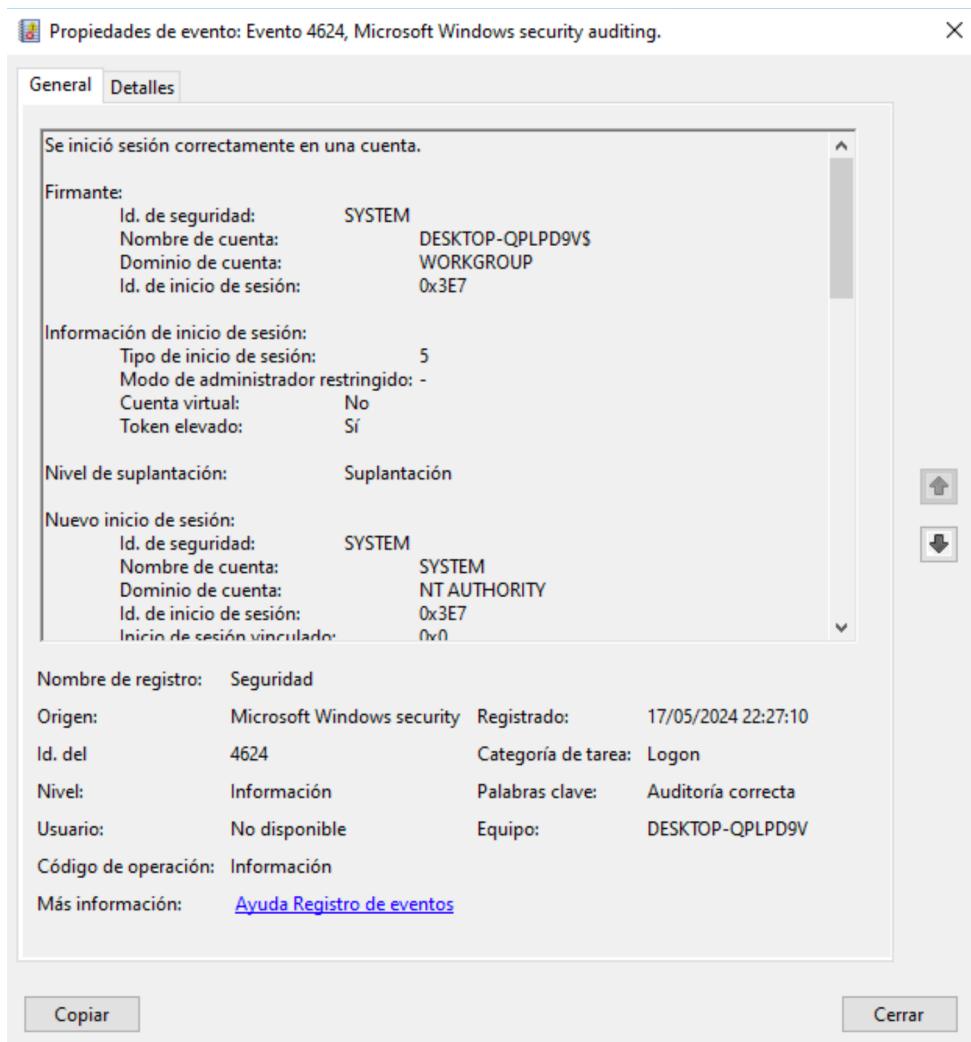
Como se ha mencionado en ejercicios anteriores, los logs o registros de eventos son muy importantes conocer los eventos que suceden en el sistema operativo, tanto a nivel de aplicación como de seguridad y sistema. En el ejercicio anterior se ha trabajado con ellos desde PowerShell, pero también es posible hacerlo a través de interfaz gráfica gracias al “Visor de Eventos” de Windows. En este ejercicio se pide obtener el “Nombre del Proceso”, el “ID del evento” y la fecha de registro de un log de categoría “Logon”. ¿Qué diferencia hay entre un log de categoría “Special Logon” con respecto a otro de categoría “Logon”?

- Interfaz Gráfica:

El Visor de Eventos de Windows es una herramienta gráfica que permite ver y analizar los registros de eventos del sistema operativo, aplicaciones y servicios. Es muy útil para la administración del sistema, la solución de problemas y la seguridad. Se puede abrir el Visor de Eventos buscando Visor de eventos en el menú de inicio de Windows.

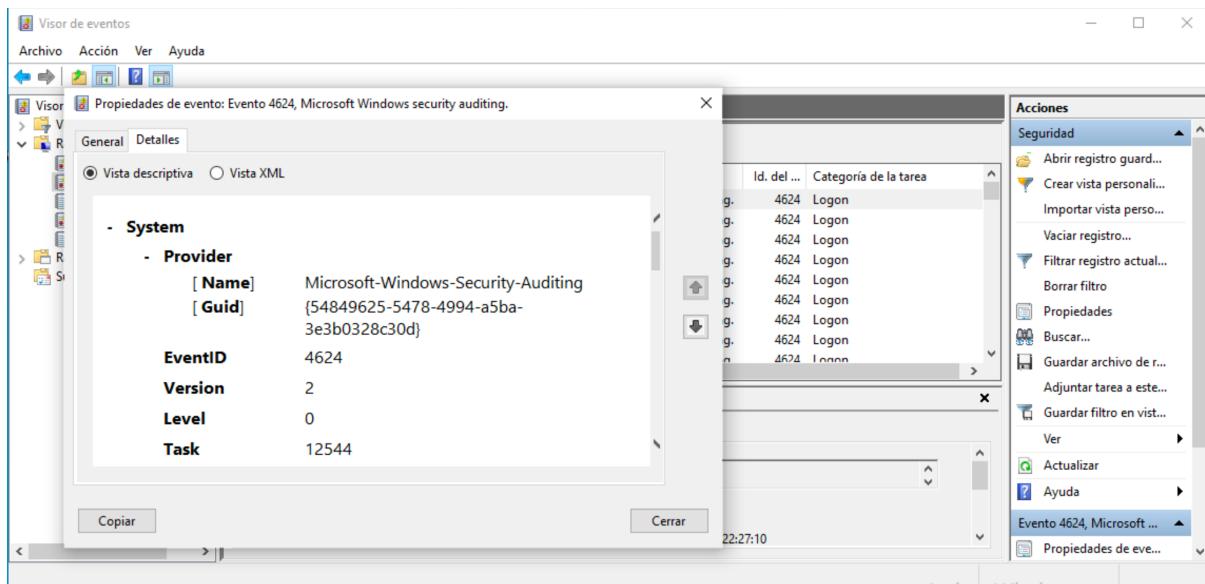


Se hace doble clic al evento y se puede visualizar la Fecha y hora del registro.

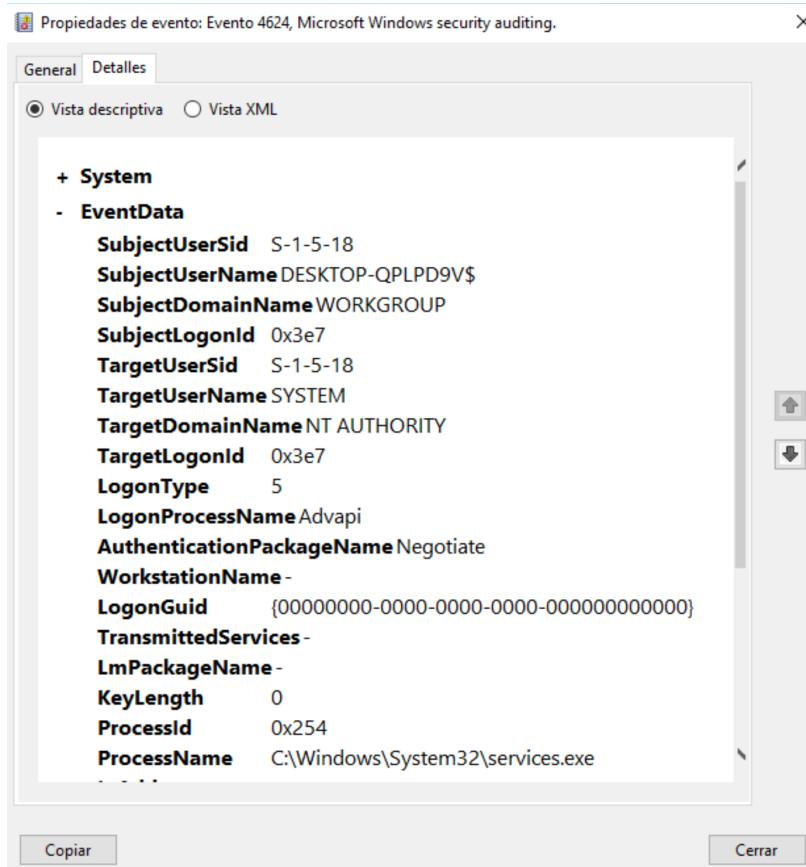


Nos dirigimos a la pestaña de detalles para la siguiente información:

- ID del evento:



- Nombre del proceso:



### Comparar "Special Logon" y "Logon"

- Log de Categoría "Logon"

Evento ID 4624: Representa un inicio de sesión exitoso. Contiene información sobre el inicio de sesión, como el usuario, el dominio, la hora del evento, el tipo de inicio de sesión, y el origen de inicio de sesión.

Propósito: Principalmente para auditar los inicios de sesión en el sistema, proporcionando detalles esenciales sobre quién inició sesión y cómo se realizó el inicio de sesión.

- Log de Categoría "Special Logon"

Evento ID 4672: Representa un inicio de sesión especial. Este tipo de evento se genera cuando una cuenta de usuario que es miembro del grupo de administradores locales inicia sesión.

Propósito: Indicar que un inicio de sesión involucra derechos administrativos. Es importante para auditar acciones de usuarios con privilegios elevados y detectar posibles actividades sospechosas por parte de administradores o usuarios con derechos especiales.

### **Ejemplo de Diferencias:**

- Logon (ID 4624):

Contexto: Generado para cualquier inicio de sesión exitoso.

Uso: Monitorear todos los inicios de sesión en el sistema.

- Special Logon (ID 4672):

Contexto: Generado cuando un usuario con privilegios especiales (como un administrador) inicia sesión.

Uso: Auditar actividades de usuarios con derechos administrativos para asegurar que no haya uso indebido de privilegios elevados.

## **Tarea 12:**

**Los firewalls o cortafuegos son dispositivos de seguridad de red que monitorizan el tráfico entrante y saliente y pueden decidir qué hacer con ese tráfico a través de reglas que se pueden definir en ellos para, por ejemplo, no permitir que entre tráfico a nuestra red proveniente de un cierto país u organización. Hay una gran variedad de restricciones que se pueden definir para permitir o bloquear cierto tráfico, ya sea saliente o entrante a nuestra red. Antes de definir una regla del firewall de Windows es interesante probar a hacer “ping” desde la CMD a la dirección IP 8.8.8.8, que pertenece a Google, para comprobar si se tiene conexión con Google (no debería dar ningún problema)**

inicialmente). “ping” es una herramienta de diagnóstico en redes que comprueba la conectividad del equipo con otro equipo remoto de una red a nivel IP.

Para ello envía paquetes ICMP de solicitud y respuesta. El funcionamiento de este protocolo se verá en el módulo de Networking, por lo que para la realización de este ejercicio basta con saber que utiliza el protocolo ICMP. Para comenzar a trabajar con el firewall de Windows en este ejercicio hay que crear una regla de salida llamada “Bloquear la conexión a Google” desde la interfaz gráfica. Para ello hay que acceder a la configuración avanzada del firewall de Windows, y dentro de “Reglas de Salida” crear dicha regla. Una vez creada, se recomienda comprobar si dicha regla se ha realizado de la manera correcta, intentando hacer “ping” a Google (a la dirección IP 8.8.8.8). Si tras comprobar si se puede hacer ping a dicha IP obtenemos que no, quiere decir que se ha creado bien, y por tanto se puede pasar a la siguiente tarea, que sería desactivar dicha regla, pero esta vez usando PowerShell. Una vez desactivada se recomienda volver a comprobar si se permite hacer “ping” a la dirección que se había bloqueado.

- cmd:

Para acceder a la línea de comandos en Windows, seguimos estos pasos:

- Hacemos clic en el botón de inicio de Windows en la esquina inferior izquierda de la pantalla.
- En el cuadro de búsqueda, escribimos “cmd” y presionamos Enter.
- Esto abrirá la ventana de la línea de comandos (cmd), donde podemos ingresar comandos para interactuar con el sistema operativo.

Comprobación de la conectividad con Google desde la línea de comandos:

Una vez que tenemos la ventana de la línea de comandos abierta, podemos comprobar la conectividad con Google utilizando el comando “ping”.

Al presionar Enter, el comando “ping” enviará paquetes de datos a la dirección de Google y recibirá respuestas. Esto nos permite verificar si podemos establecer una conexión con el servidor de Google y si hay una conectividad de red adecuada desde nuestro sistema.

```
C:\ Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.4412]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\soced>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=24ms TTL=127
Respuesta desde 8.8.8.8: bytes=32 tiempo=24ms TTL=127
Respuesta desde 8.8.8.8: bytes=32 tiempo=25ms TTL=127
Respuesta desde 8.8.8.8: bytes=32 tiempo=24ms TTL=127

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 24ms, Máximo = 25ms, Media = 24ms

C:\Users\soced>
```

- Interfaz gráfica:

Para crear una nueva regla en el Firewall de Windows Defender a través de la interfaz gráfica, seguimos estos pasos:

- Abrimos el Panel de Control desde el menú de inicio.
- Dentro del Panel de Control, seleccionamos "Sistema y Seguridad".
- Luego, hacemos clic en "Firewall de Windows Defender" para acceder a la configuración del firewall.
- En la ventana del Firewall de Windows Defender, elegimos la opción "Configuración avanzada" en el panel izquierdo.
- Esto abrirá la ventana de "Reglas de salida", donde podemos administrar las reglas salientes del firewall.
- Para crear una nueva regla, hacemos clic en "Nueva regla" en el panel derecho.
- A continuación, seguimos las instrucciones del asistente para crear la regla según nuestras necesidades específicas.

**Firewall de Windows Defender**

Ventana principal del Panel de control

Ayudar a proteger el equipo con Firewall de Windows Defender

Firewall de Windows Defender puede ayudar a impedir que piratas informáticos o software malintencionado obtengan acceso al equipo a través de Internet o una red.

Redes privadas Conectado

Redes domésticas o del trabajo en cuyos usuarios y dispositivos confíe

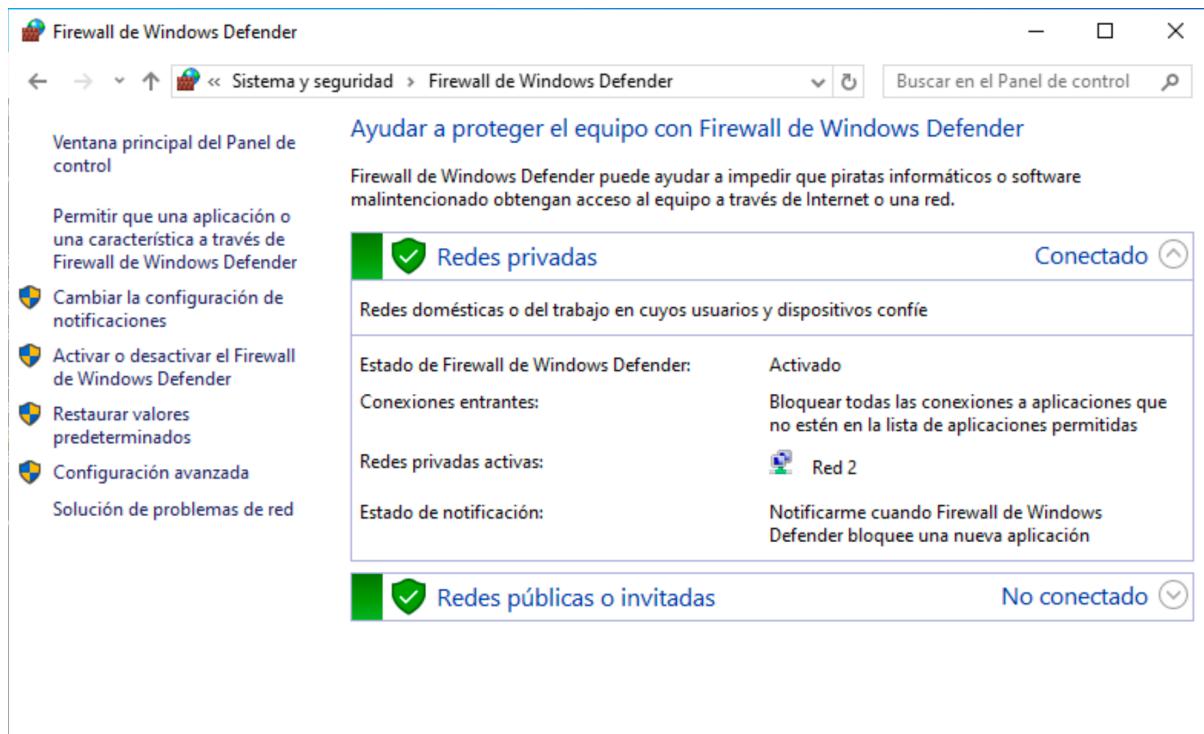
Estado de Firewall de Windows Defender: Activado

Conexiones entrantes: Bloquear todas las conexiones a aplicaciones que no estén en la lista de aplicaciones permitidas

Redes privadas activas: Red 2

Estado de notificación: Notificarme cuando Firewall de Windows Defender bloquee una nueva aplicación

Redes públicas o invitadas No conectado



**Windows Defender Firewall con seguridad avanzada**

Archivo Acción Ver Ayuda

Windows Defender Firewall con seguridad avanzada en Equipo local

Información general

**Perfil de dominio**

- Firewall de Windows Defender está activado.
- Las conexiones entrantes que no coincidan con una regla están bloqueadas.
- Conexiones salientes que no coincidan con una regla serán permitidas.

**El perfil privado está activo.**

- Firewall de Windows Defender está activado.
- Las conexiones entrantes que no coincidan con una regla están bloqueadas.
- Conexiones salientes que no coincidan con una regla serán permitidas.

**Perfil público**

- Firewall de Windows Defender está activado.
- Las conexiones entrantes que no coincidan con una regla están bloqueadas.
- Conexiones salientes que no coincidan con una regla serán permitidas.

Propiedades de Firewall de Windows Defender

Introducción

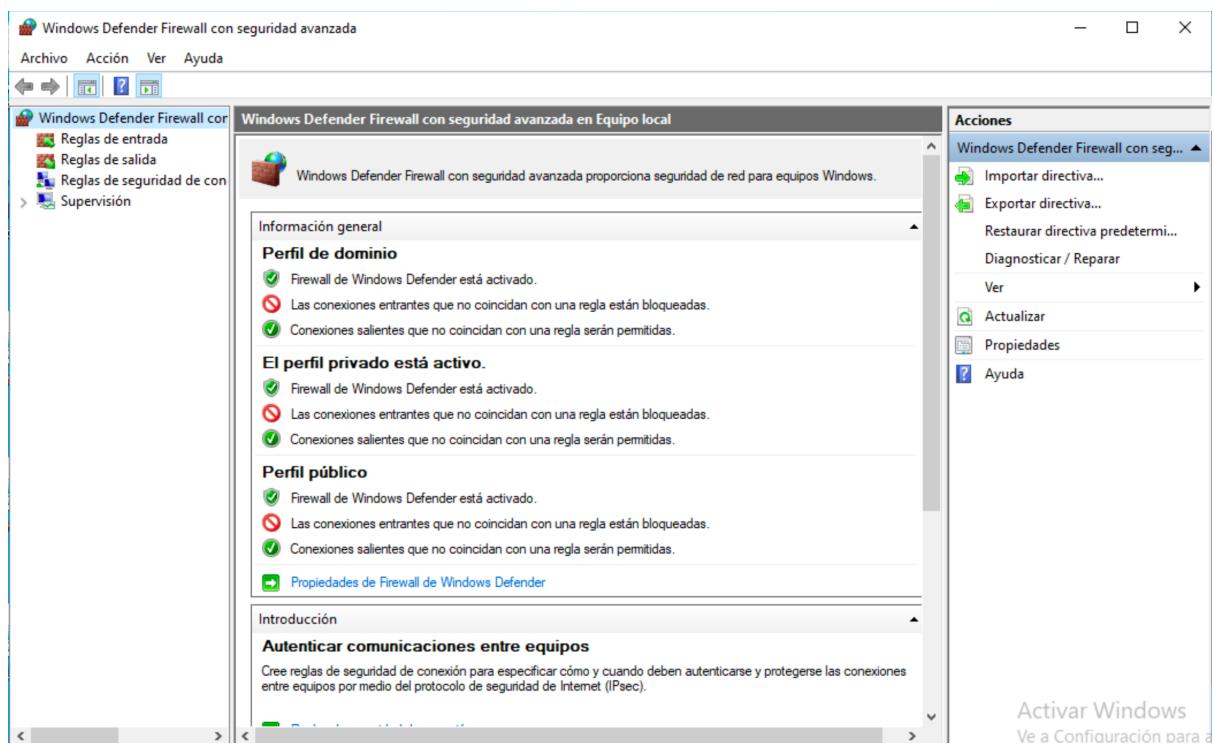
**Autenticar comunicaciones entre equipos**

Cree reglas de seguridad de conexión para especificar cómo y cuándo deben autenticarse y protegerse las conexiones entre equipos por medio del protocolo de seguridad de Internet (IPsec).

Acciones

- Importar directiva...
- Exportar directiva...
- Restaurar directiva predeterminada
- Diagnosticar / Reparar
- Ver
- Actualizar
- Propiedades
- Ayuda

Activar Windows  
Ve a Configuración para a



**Windows Defender Firewall con seguridad avanzada**

Archivo Acción Ver Ayuda

Windows Defender Firewall con seguridad avanzada en Equipo local

Información general

**Perfil de dominio**

- Firewall de Windows Defender está activado.
- Las conexiones entrantes que no coincidan con una regla están bloqueadas.
- Conexiones salientes que no coincidan con una regla serán permitidas.

**El perfil privado está activo.**

- Firewall de Windows Defender está activado.
- Las conexiones entrantes que no coincidan con una regla están bloqueadas.
- Conexiones salientes que no coincidan con una regla serán permitidas.

**Perfil público**

- Firewall de Windows Defender está activado.
- Las conexiones entrantes que no coincidan con una regla están bloqueadas.
- Conexiones salientes que no coincidan con una regla serán permitidas.

[Propiedades de Firewall de Windows Defender](#)

Introducción

**Autenticar comunicaciones entre equipos**

Cree reglas de seguridad de conexión para especificar cómo y cuando deben autenticarse y protegerse las conexiones entre equipos por medio del protocolo de seguridad de Internet (IPsec).

Acciones

- Importar directiva...
- Exportar directiva...
- Restaurar directiva predeterminada...
- Diagnosticar / Reparar
- Ver
- Actualizar
- Propiedades
- Ayuda

Activar Windows

Ve a Configuración para activar

---

**Asistente para nueva regla de salida**

Archivo Acción Ver Ayuda

Tipo de regla

Seleccione el tipo de regla de firewall que desea crear.

Pasos:

- Tipo de regla
- Programa
- Protocolo y puertos
- Ámbito
- Acción
- Perfil
- Nombre

¿Qué tipo de regla desea crear?

**Programa**  
Regla que controla las conexiones de un programa.

**Puerto**  
Regla que controla las conexiones de un puerto TCP o UDP.

**Predefinida:**  
Regla que controla las conexiones de una experiencia con Windows.

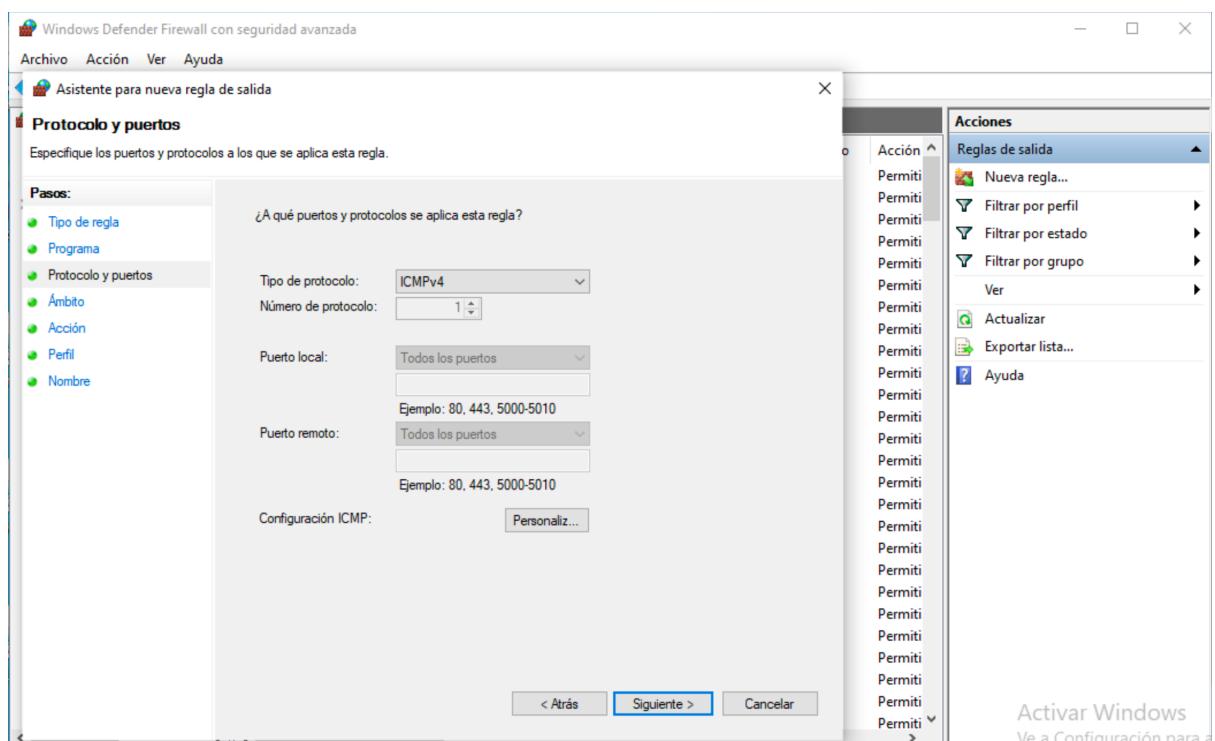
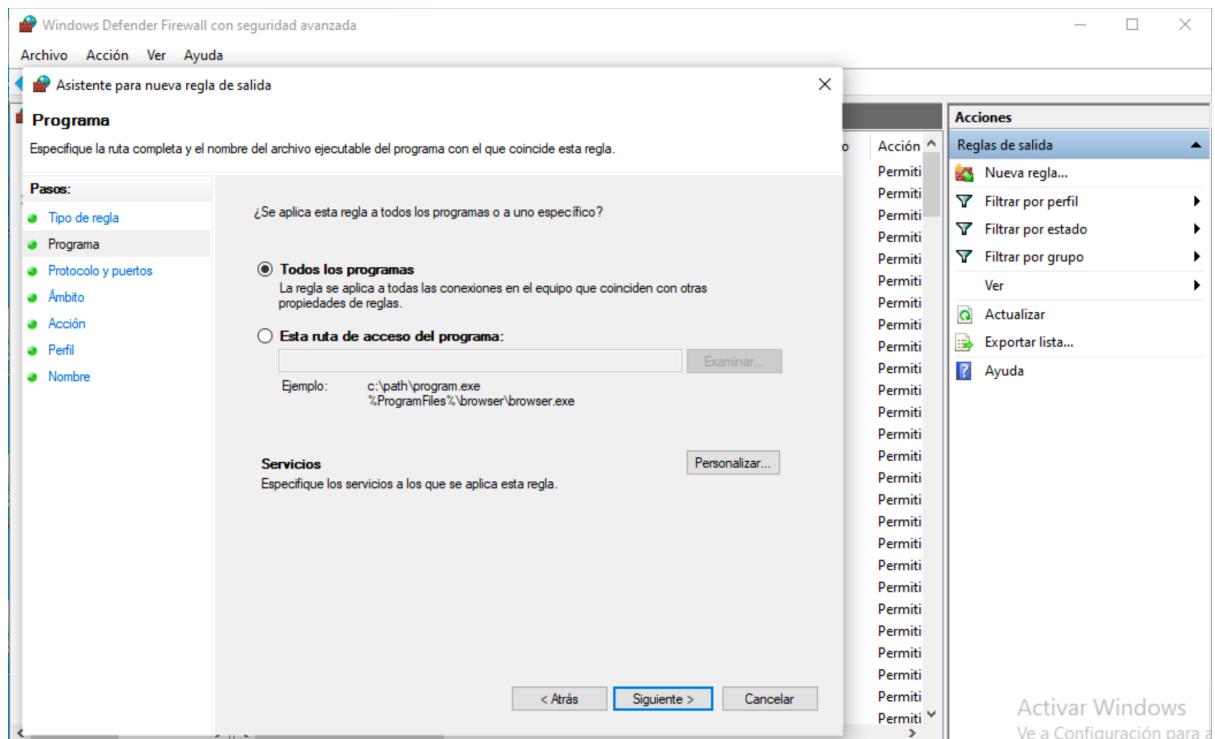
**Personalizada**  
Regla personalizada.

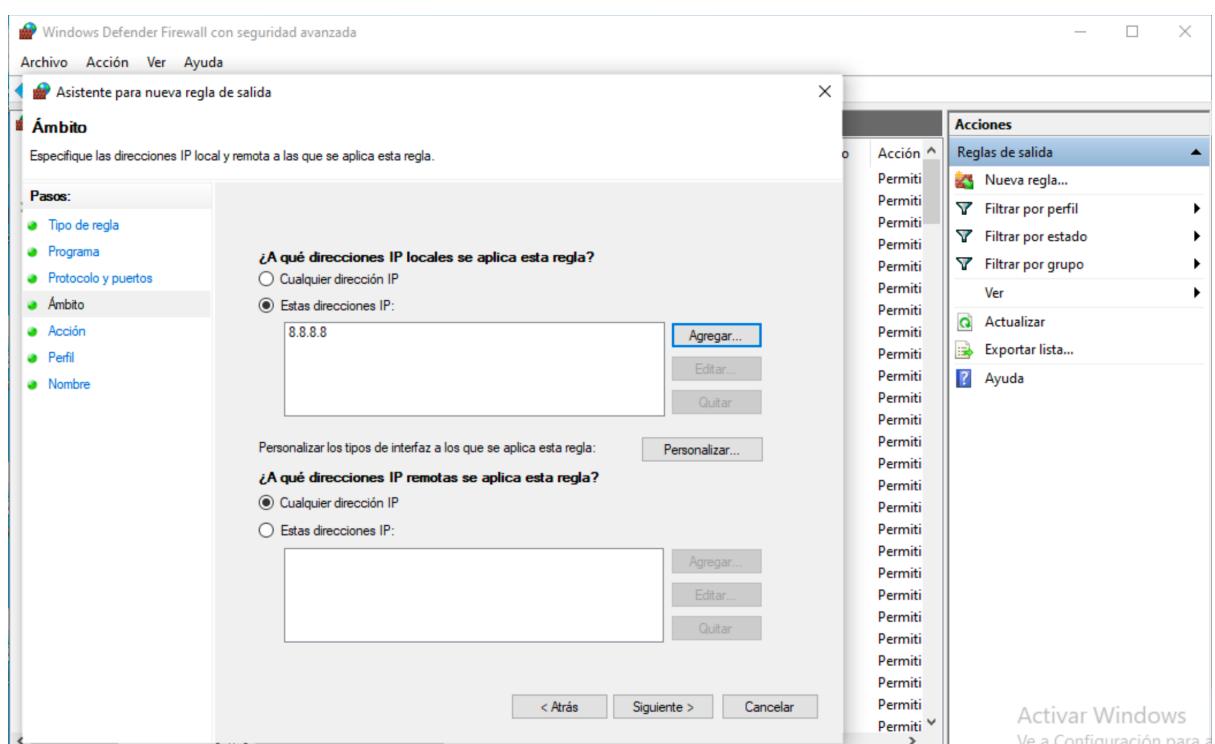
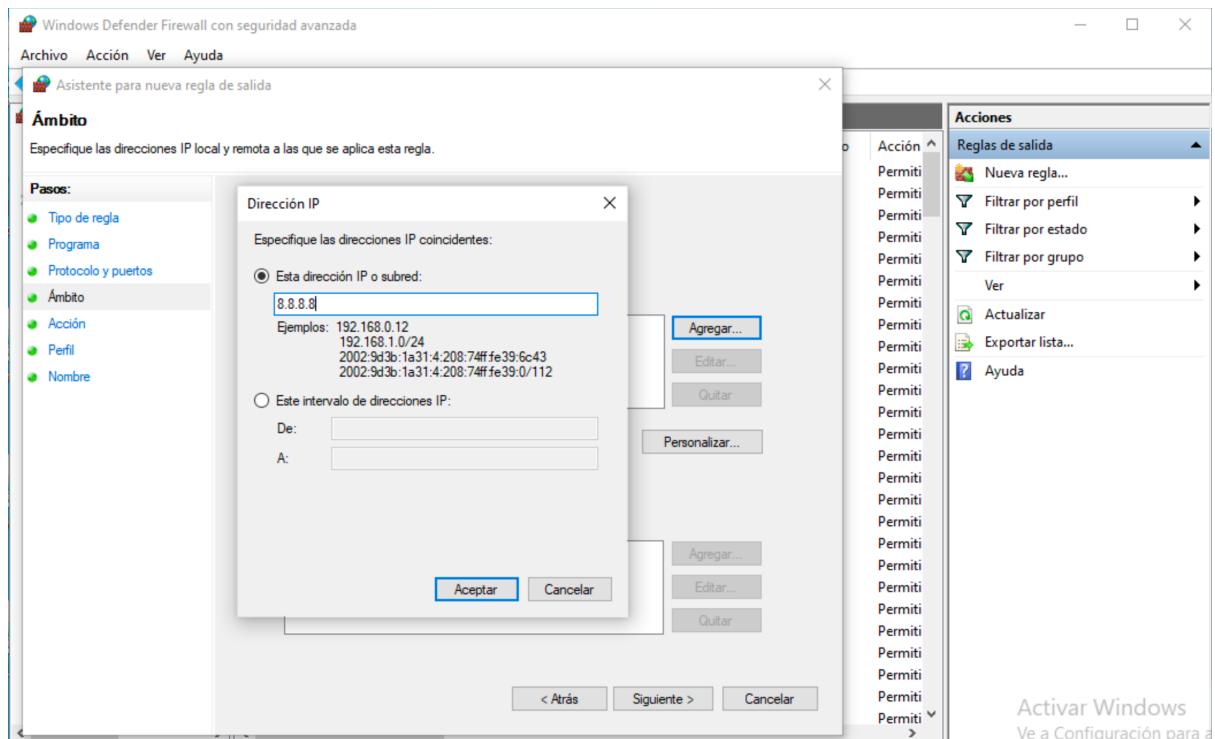
Acciones

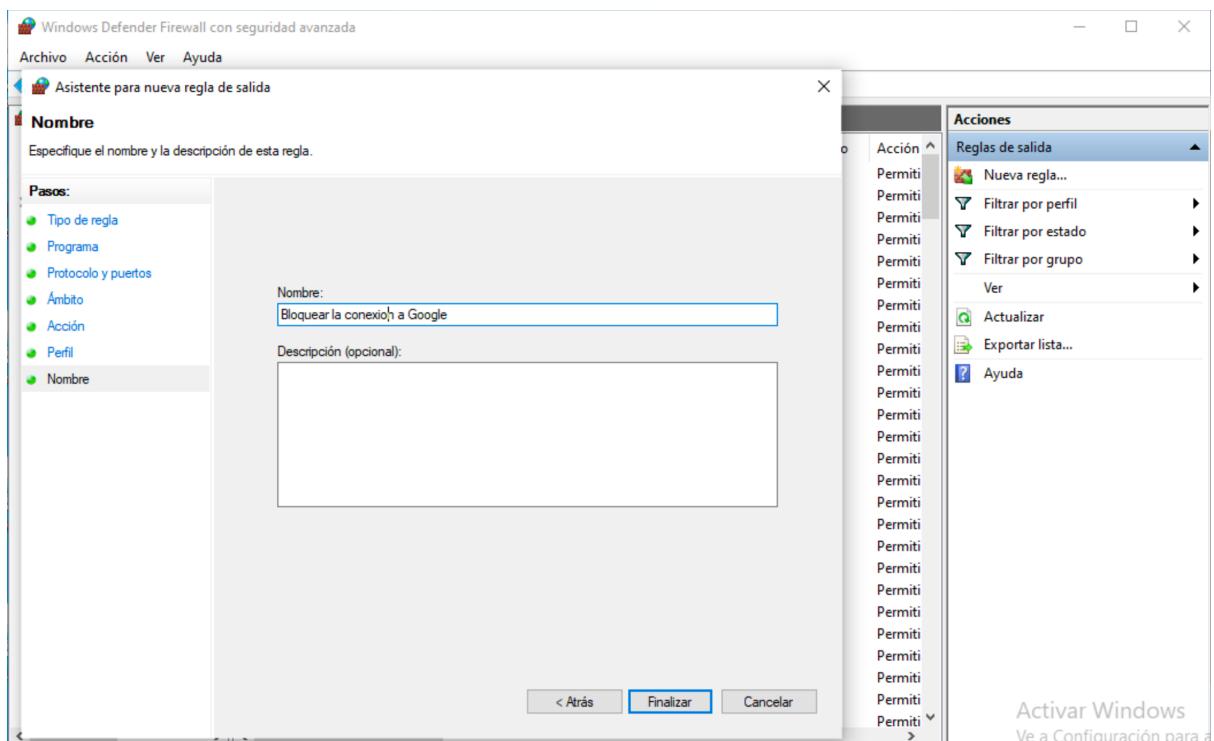
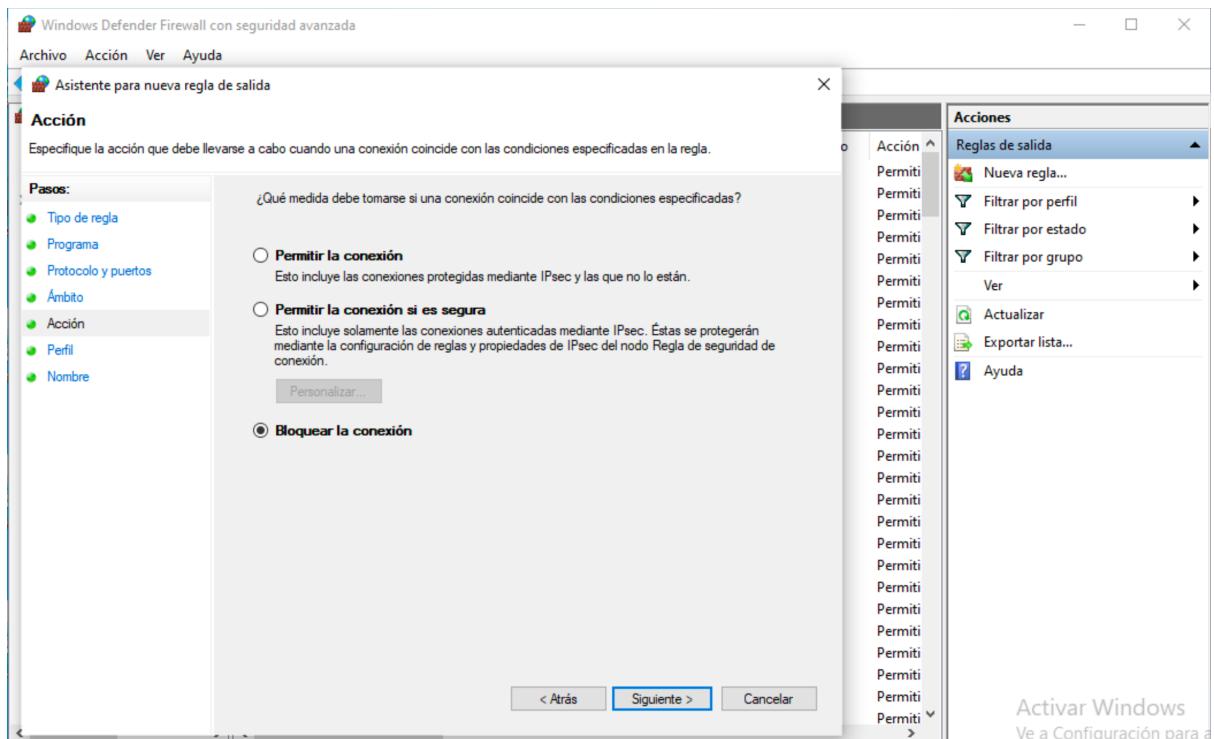
- Reglas de salida
- Nueva regla...
- Filtrar por perfil
- Filtrar por estado
- Filtrar por grupo
- Ver
- Actualizar
- Exportar lista...
- Ayuda

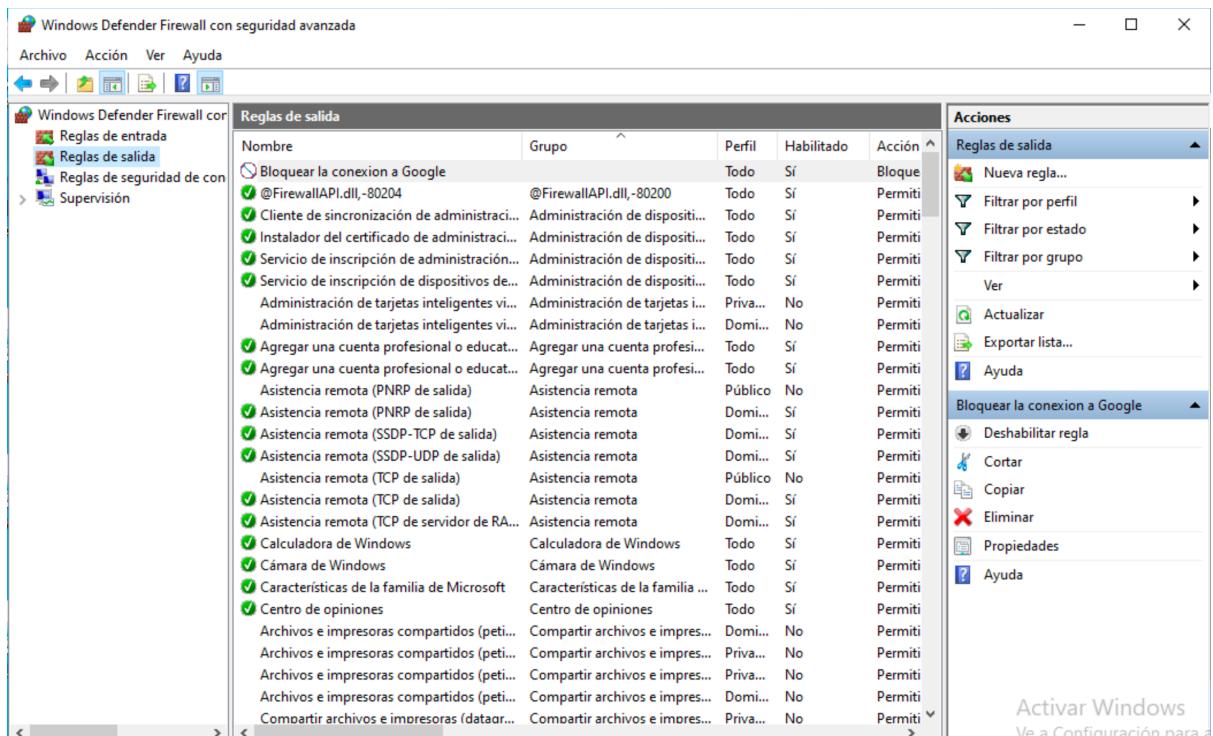
Activar Windows

Ve a Configuración para activar









- PowerShell:

Para desactivar la regla se puede utilizar la opción deshabilitar regla o ejecutar en PowerShell:

```
Disable-NetFirewallRule -DisplayName "Bloquear la conexión a Google"
```

## Tarea 13:

**A la hora de crear contraseñas en nuestra vida cotidiana es muy importante crearlas de manera que sean seguras, evitando patrones sencillos, datos personales, etc., para estar más protegidos ante los ciberdelincuentes, ya que una contraseña débil facilita a estos para obtener acceso a nuestras cuentas. Si en nuestra vida cotidiana es importante una contraseña segura, en el mundo profesional lo es aún más, ya que, si un atacante consigue acceso a nuestra cuenta o equipo, pone en riesgo los datos de los compañeros, de la empresa e incluso de los clientes que esta empresa pueda tener. Por tanto, es imprescindible utilizar contraseñas seguras, de una larga longitud y aleatorios. Para ello se usan aplicaciones, las cuales pueden ser en la nube, que tienen el problema de que dependen de terceros y pueden sufrir una filtración de datos, quedando nuestras contraseñas expuestas; y aplicaciones que se instalan en el propio dispositivo. Esta es una solución más segura para poder crear y almacenar todas las contraseñas en un mismo lugar, y no tener que memorizar cada una de ellas, ya que para acceder a esta base de datos únicamente habrá que recordar la “clave maestra”. El gestor de contraseñas de este tipo más conocido es KeePass. Se pide que se instale esta aplicación y, una vez instalada, generada la base de datos y creada la “clave maestra” (esta sí que hay que recordarla),**

**crear una contraseña segura (investigar sobre las opciones que da para crear las contraseñas) y guardarla.**

Este ejercicio consiste en instalar y configurar KeePass, un gestor de contraseñas ampliamente reconocido por su capacidad para generar y almacenar contraseñas seguras. A diferencia de confiar en soluciones en la nube, como se mencionó en el enunciado, KeePass nos ofrece la ventaja de tener un control total sobre nuestras contraseñas al instalar la aplicación en nuestro propio dispositivo.

Para comenzar, se procedió a descargar e instalar KeePass siguiendo los pasos indicados en las capturas de pantalla adjuntas. Una vez completada la instalación, el siguiente paso crucial fue crear una base de datos segura donde almacenar todas nuestras contraseñas. Esto garantiza que todas las credenciales estén centralizadas y protegidas bajo una sola contraseña maestra, simplificando así la gestión de contraseñas y aumentando la seguridad.

La elección de una contraseña maestra fuerte es fundamental. Debe ser única, larga y compleja para resistir los intentos de fuerza bruta por parte de los ciberdelincuentes. Una vez establecida la base de datos y definida la contraseña maestra, KeePass ofrece la opción de generar contraseñas seguras para nuestras cuentas individuales.

La herramienta de generación de contraseñas de KeePass, nos permite personalizar la longitud, complejidad y tipos de caracteres de las contraseñas generadas. Esta capacidad nos permite adaptar nuestras contraseñas a los requisitos de seguridad específicos de cada cuenta, asegurando así una protección óptima.

En resumen, la instalación y configuración de KeePass nos brinda una solución sólida y segura para la gestión de contraseñas en el entorno profesional.

