

# WINDOWS POWERSHELL

 [Descargar PDF](#)

## ▼ WINDOWS POWERSHELL

- NETTCPIP

### ▼ INTERFACES DE RED

- MOSTRAR CONFIGURACIÓN DE INTERFACES
- MOSTRAR CONFIGURACIÓN DE UN INTERFAZ
- MOSTRAR CONFIGURACIÓN DNS
- ACTIVAR Y DESACTIVAR INTERFACES
- CAMBIAR NOMBRE A UNA INTERFAZ

### ▼ CONFIGURACIÓN ESTÁTICA

- CONFIGURAR IP Y GATEWAY
- CONFIGURAR DNS
- BORRAR IP Y GATEWAY
- BORRAR DNS

### ▼ CONFIGURACIÓN DINÁMICA

- CONFIGURAR IP Y GATEWAY
- CONFIGURAR DNS
- LIBERAR Y RENOVAR IP DINÁMICA

## ▼ TABLA DE ENRUTAMIENTO

- MOSTRAR TABLA

### ▼ MODIFICAR TABLA

- AÑADIR ENTRADA
- BORRAR ENTRADA
- MODIFICAR ENTRADA

## ▼ TABLA ARP

- MOSTRAR TABLA

### ▼ MODIFICAR TABLA

- AÑADIR ENTRADA
- BORRAR ENTRADA

## ■ INTERFACES DE RED INÁLAMBRICAS

# NETTCPIP

La mayor parte de los cmdlets que se utilizan para la configuración de red pertenecen al módulo **NetTCPIP**. En la mayoría de los cmdlets de NetTCPIP se puede usar indistintamente el nombre descriptivo de la interfaz ( `-Name` o `-InterfaceAlias` ) o su número de índice ( `-InterfaceIndex` )

## INTERFACES DE RED

### MOSTRAR CONFIGURACIÓN DE INTERFACES

Podemos mostrar la configuración de todos los adaptadores de red con el comando [Get-NetAdapter](#) de PowerShell.

#### Mostrar interfaces de red

Mostrar el nombre corto, nombre descriptivo e índice de todos los interfaces de red.

```
PS C:\>Get-NetAdapter | Format-Table -Property Name, InterfaceDescription, InterfaceIndex
Name           InterfaceDescription          InterfaceIndex
---           -----
MiRed          Realtek PCIe GbE Family Controller      18
vEthernet (Default Switch)  Hyper-V Virtual Ethernet Adapter    35
RedVirtualBox   VirtualBox Host-Only Ethernet Adapter     5
```

### MOSTRAR CONFIGURACIÓN DE UN INTERFAZ

También se puede realizar con el comando [Get-NetAdapter](#).

```
PS C:\>Get-NetAdapter -name <IFaceName> | Format-List
```

#### Mostrar información de una interfaz de red

Mostrar toda la información de la interfaz de red de nombre *MiRed*.

```
C:\>Get-NetAdapter -name MiRed | Format-List
Name           : RedAula
InterfaceDescription : Realtek PCIe GbE Family Controller
InterfaceIndex   : 18
MacAddress       : 40-8D-5C-2A-2B-A5
MediaType        : 802.3
PhysicalMediaType : 802.3
InterfaceOperationalStatus : Up
AdminStatus       : Up
LinkSpeed(Mbps)   : 100
MediaConnectionState : Connected
ConnectorPresent  : True
DriverInformation : Driver Date 2021-05-11 Version 10.50.511.2021 NDIS 6.40
```

También se puede obtener información adicional con los cmdlets [Get-NetIPAddress](#) y [Get-NetIPInterface](#).

```
PS C:\>Get-NetIPAddress [-InterfaceAlias <InterfaceName>] | Format-List  
PS C:\>Get-NetIPInterface [-InterfaceAlias <InterfaceName>] | Format-List
```

Podemos obtener datos adicionales como la IP, la máscara o la IP del servidor DHCP con el cmdlet [Get-WmiObject](#) que actúa sobre objetos [WMI](#) (*Windows Management Instrumentation*):

```
PS C:\>(Get-WmiObject Win32_NetworkAdapterConfiguration)[1].IPAddress[0]  
PS C:\>(Get-WmiObject Win32_NetworkAdapterConfiguration)[1].IPSubnet[0]  
PS C:\>(Get-WmiObject Win32_NetworkAdapterConfiguration)[1].DHCPServer
```

## MOSTRAR CONFIGURACIÓN DNS

Se utiliza el cmdlet [Get-DnsClientServerAddress](#) de PowerShell.

```
PS C:\>Get-DnsClientServerAddress [-InterfaceAlias <InterfaceName>]
```

Si no se introduce la opción `-InterfaceAlias` entonces muestra la configuración de todas las interfaces.

### Configuración cliente DNS

Mostrar la configuración del DNS asociada al adaptador de red de nombre *MiRed*.

```
C:\>Get-DnsClientServerAddress -InterfaceAlias MiRed | Format-Table -AutoSize  
InterfaceAlias InterfaceIndex AddressFamily ServerAddresses  
-----  
MiRed 6 IPv4 {1.1.1.1, 4.4.4.4}  
MiRed 6 IPv6 {}
```

## ACTIVAR Y DESACTIVAR INTERFACES

Para activar y desactivar interfaces se utilizan los comandos de PowerShell [Enable-NetAdapter](#) y [Disable-NetAdapter](#) respectivamente.

```
PS C:\>Disable-NetAdapter -name <InterfaceName> [-Confirm:$false]  
PS C:\>Enable-NetAdapter -name <InterfaceName> [-Confirm:$false]
```

### Desactivar interfaces de red

Desactiva la interfaz de nombre *MiRed*.

```
PS C:\>Disable-NetAdapter -name MiRed -Confirm:$false
```

# CAMBIAR NOMBRE A UNA INTERFAZ

Se utiliza el comando [Rename-NetAdapter](#) de PowerShell.

```
C:\>Rename-NetAdapter -name <IFaceName> -NewName <NewIFaceName>
```

## Cambiar nombre de interfaz de red

Cambiar el nombre de la interfaz *Ethernet* por *MiRed*.

```
PS C:\>Rename-NetAdapter -Name Ethernet -NewName MiRed
```

# CONFIGURACIÓN ESTÁTICA

## CONFIGURAR IP Y GATEWAY

El comando [New-NetIPAddress](#) se utiliza para asignar una IP a una interfaz de red así como para configurar la puerta de enlace.

```
PS C:\>New-NetIPAddress -InterfaceAlias <IFaceName> -IPAddress <IP> [-PrefixLength <bits-xarxa>] [-DefaultGateway <IP-Gateway>] `>> [-PolicyStore {ActiveStore,PersistentStore}] [-Confirm:$false]
```

Hay que tener en cuenta que:

- Una interfaz de red puede tener mas de una IP. Por ejemplo, si hacemos dos veces [New-NetIPAddress](#) con direcciones IP diferentes tendremos dos configuraciones para la interfaz.
- Si no se indica valor para `-PolicyStore` entonces se da de alta como persistente y activa.

## Configurar IP estáticamente

Configurar la interfaz *MiRed* con la ip de clase C `192.168.1.20` y con la puerta de enlace `192.168.1.100`.

```
PS C:\>New-NetIPAddress -InterfaceAlias MiRed -IPAddress 192.168.1.20 `>> -PrefixLength 24 -DefaultGateway 192.168.1.100 -Confirm:$false
```

Como no hemos utilizado el parámetro `-PolicyStore` se dará de alta por defecto como persistente y activa.

## CONFIGURAR DNS

Con el comando [Set-DnsClientServerAddress](#) se configura el cliente DNS (o clientes) al que inicialmente se envían las peticiones de resolución DNS.

```
PS C:\>Set-DnsClientServerAddress -InterfaceAlias <IFaceName> `>> -ServerAddresses <IP_Server1,IP_Server2> [-Validate:$false] [-Confirm:$false]
```

## Configuración cliente DNS

Configurar la interfaz de red *MiRed* con los DNS de IP `1.1.1.1` y `8.8.8.8`.

```
PS C:\>Set-DnsClientServerAddress -InterfaceAlias MiRed -ServerAddresses `  
  >> 1.1.1.1, 8.8.8.8 -Validate:$false
```

El campo `-validate:$false` permite que la orden no intente conectar con el servidor para comprobar que existe.

## BORRAR IP Y GATEWAY

Para borrar la configuración de la IP y/o puerta de enlace de un adaptador de red utilizamos el comando [Remove-NetIPAddress](#) de PowerShell.

```
PS C:\>Remove-NetIPAddress -InterfaceAlias <IFaceName> -IPAddress <IP> `  
  >> -DefaultGateway <IP-Gateway> [-Confirm:$false]
```

### Borrar ip estática

Elimina la ip `192.18.1.20` y la puerta de enlace de la interfaz de red *MiRed*.

```
C:\>Remove-NetIPAddress -InterfaceAlias MiRed -IPAddress 192.168.1.20 `  
  >> -DefaultGateway 192.168.1.100 -Confirm:$false
```

## BORRAR DNS

Para borrar la configuración del DNS de un adaptador de red utilizamos el comando [Set-DnsClientServerAddress](#) de PowerShell.

```
PS C:\>Set-DnsClientServerAddress -InterfaceAlias <IFaceName> -ResetServerAddresses [-Confirm:$false]
```

### Borrar DNS de IP estática

Borrar la configuración dns para la interfaz *MiRed*.

```
PS C:\>Set-DnsClientServerAddress -InterfaceAlias MiRed -ResetServerAddresses -Confirm:$false
```

# CONFIGURACIÓN DINÁMICA

Si previamente se ha realizado una configuración estática, es recomendable eliminarla antes de aplicar una configuración dinámica.

## CONFIGURAR IP Y GATEWAY

Para configurar una interfaz de red para que obtenga su dirección IP y puerta de enlace mediante DHCP se utiliza el comando [Set-NetIPInterface](#).

```
PS C:\>Set-NetIPInterface -InterfaceAlias <IFaceName> -dhcp Enabled
```

### Configurar IP dinámica

Configurar la interfaz de red *MiRed* para que obtenga su dirección IP y puerta de enlace mediante DHCP.

```
PS C:\>Set-NetIPInterface -InterfaceAlias MiRed -dhcp Enabled
```

Esta configuración es persistente mientras dure el tiempo de alquiler (*lease time*) del DHCP.

## CONFIGURAR DNS

No es necesario configurarlo. Al configurar la interfaz como dhcp el sistema obtendrá el servidor dns a través de dicho protocolo.

## LIBERAR Y RENOVAR IP DINÁMICA

Aunque es más fácil hacerlo mediante `ipconfig /release` o `ipconfig /renew` tambien se puede realizar mediante objetos de powershell.

El proceso es el siguiente:

- **Obtener el objeto WMI para tratar la configuración**

```
PS C:\>$MyInterfaceIndex=(Get-NetAdapter -InterfaceAlias <IFaceName> | Format-Wide -Property InterfaceIndex | Out-String).Trim()  
PS C:\>$MyLan=Get-WmiObject Win32_NetworkAdapterConfiguration | where-object ($_.Index -match $MyInterfaceIndex)
```

- **Liberar la IP obtenida por dhcp**

```
PS C:\>$MyLan.ReleaseDHCPLease()
```

- **Renovar la IP obtenida por dhcp**

```
PS C:\>$MyLan.RenewDHCPLease()
```

# TABLA DE ENRUTAMIENTO

## MOSTRAR TABLA

Para mostrar la tabla de enrutamiento utilizamos el comando [Get-NetRoute](#) de PowerShell.

```
PS C:\>Get-NetRoute [-AddressFamily {IPv4,IPv6}]
```

Si no se indica `-AddressFamily` muestra las entradas para IPv4 e IPv6.

### Mostrar tabla de enrutamiento

Mostrar la tabla de enrutamiento para las direcciones de la familia IPv4.

```
PS C:\>Get-NetRoute -AddressFamily IPV4
ifIndex DestinationPrefix     NextHop      RouteMetric ifMetric PolicyStore
----- -----
6       255.255.255.255/32  0.0.0.0      256          25        ActiveStore
1       255.255.255.255/32  0.0.0.0      256          25        ActiveStore
6       224.0.0.0/4          0.0.0.0      256          75        ActiveStore
1       127.255.255.255/32  0.0.0.0      256          75        ActiveStore
1       127.0.0.1/32         0.0.0.0      256          75        ActiveStore
1       127.0.0.0/8          0.0.0.0      256          75        ActiveStore
6       10.0.2.255/32        0.0.0.0      256          25        ActiveStore
6       10.0.2.15/32         0.0.0.0      256          25        ActiveStore
6       10.0.2.0/24          0.0.0.0      256          25        ActiveStore
6       0.0.0.0/0             10.0.2.2    0            25        ActiveStore
```

## MODIFICAR TABLA

### AÑADIR ENTRADA

Para añadir una ruta estática a la tabla de enrutamiento utilizamos el comando [New-NetRoute](#) de PowerShell.

```
PS C:\>New-NetRoute -DestinationPrefix <"IP/prefix"> ` 
[-Nexthop <IP-Gateway>] ` 
-InterfaceAlias <IFaceName> ` 
[-PolicyStore {ActiveStore, PersistentStore}] ` 
[-Confirm:$false]
```

### Añadir ruta estática

Añadir una entrada a la tabla de enrutamiento para acceder a la red `192.168.2.0/24` a través de la puerta de enlace de ip `192.168.1.100` por la interfaz *MiRed*.

```
PS C:\>New-NetRoute -DestinationPrefix "192.168.2.0/24" -NextHop 192.168.1.100 ` 
-InterfaceAlias MiRed -Confirm:$false
```

## Añadir ruta por defecto

Añadir una entrada a la tabla de enrutamiento para acceder a cualquier red (ruta por defecto) a través de la puerta de enlace de ip `192.168.1.200` por la interfaz *MiRed*.

```
PS C:\>New-NetRoute -DestinationPrefix "0.0.0.0/0" -NextHop 192.168.1.200 `  
-InterfaceAlias MiRed -Confirm:$false
```

## BORRAR ENTRADA

Para eliminar una ruta estática a la tabla de enrutamiento utilizamos el comando `Remove-NetRoute` de PowerShell.

```
PS C:\>Remove-NetRoute -DestinationPrefix <"IP/prefix"> `  
[-Nexthop <IP-Gateway>] `  
-InterfaceAlias <IFaceName> `  
[-PolicyStore {ActiveStore, PersistentStore}] `  
[-Confirm:$false]
```

## Eliminar ruta estática

Borrar la entrada para la ruta `192.168.2.0/24` con gateway `192.168.1.100` que tiene salida por la interfaz *MiRed*.

```
C:\>Remove-NetRoute -DestinationPrefix "192.168.2.0/24" -NextHop 192.168.1.100 `  
-InterfaceAlias MiRed -Confirm:$false
```

## MODIFICAR ENTRADA

Aunque se puede utilizar el cmdlet `Set-NetRoute` es mejor primero borrar la entrada y después volver a añadirla.

Evidentemente para todos estos cmdlets si utilizamos `-PolicyStore ActiveStore` la configuración será no persistente.

## TABLA ARP

### MOSTRAR TABLA

Para mostrar el contenido de la tabla ARP utilizamos el comando `Get-NetNeighbor`.

```
PS C:\>Get-NetNeighbor [-InterfaceAlias <IFaceName>] [-AddressFamily {IPv4,IPv6}]
```

Si no se indica interfaz entonces muestra la tabla ARP de todas las interfaces del sistema.

## Mostrar tabla ARP

Mostrar la tabla ARP de la interfaz de nombre *MiRed*.

ifIndex	IPAddress	LinkLayerAddress	State	PolicyStore
6	255.255.255.255	FF-FF-FF-FF-FF-FF	Permanent	ActiveStore
6	239.255.255.250	01-00-5E-7F-FF-FA	Permanent	ActiveStore
6	224.0.0.252	01-00-5E-00-00-FC	Permanent	ActiveStore
6	224.0.0.251	01-00-5E-00-00-FB	Permanent	ActiveStore
6	224.0.0.22	01-00-5E-00-00-16	Permanent	ActiveStore
6	10.0.2.255	FF-FF-FF-FF-FF-FF	Permanent	ActiveStore
6	10.0.2.2	52-54-00-12-35-02	Stale	ActiveStore
6	10.0.2.1	00-00-00-00-00-00	Unreachable	ActiveStore
6	10.0.2.0	00-00-00-00-00-00	Unreachable	ActiveStore

## MODIFICAR TABLA

### AÑADIR ENTRADA

Para añadir una entrada a la tabla ARP que asocia una IP a una MAC en una interfaz de red utilizamos el comando [New-NetNeighbor](#).

```
PS C:\>New-NetNeighbor -InterfaceAlias <IFaceName> -IPAddress <IP> `  
-LinkLayerAddress <MAC> [-PolicyStore {ActiveStore,PersistentStore}] [-Confirm:$false]
```

#### Añadir entrada a tabla ARP

Añadir a la tabla ARP de la interfaz MiRed la entrada que asocia la IP `192.168.1.100` a la MAC `02-00-00-00-00-01`.

```
PS C:\>New-NetNeighbor -InterfaceAlias MiRed -IPAddress 192.168.1.100 `  
-LinkLayerAddress 02-00-00-00-00-01 -Confirm:$false
```

### BORRAR ENTRADA

El comando [Remove-NetNeighbor](#) se utiliza para eliminar una o más entradas de la tabla ARP.

```
PS C:\> Remove-NetNeighbor -InterfaceAlias <IFaceName> -IPAddress <IP> `  
-LinkLayerAddress <MAC> [-PolicyStore {ActiveStore,PersistentStore}] `  
[-Confirm:$false]
```

#### Eliminar entrada de la tabla ARP

Borrar de la interfaz *MiRed* todas las entradas de la tabla ARP.

```
PS C:\> Remove-NetNeighbor -InterfaceAlias MiRed -Confirm:$false
```

Borrar de la interfaz *MiRed* la entrada para la dirección `192.168.1.100`.

```
PS C:\> Remove-NetNeighbor -InterfaceAlias MiRed -IPAddress 192.168.1.100 -Confirm:$false
```

# INTERFACES DE RED INÁLAMBRICAS

Las interfaces de red inalámbricas se configuran en el nivel de red igual que las alámbricas. Sin embargo, para conectar una interfaz inalámbrica con un punto de acceso es necesario configurar su nivel de enlace, es decir: el SSID, la clave, el algoritmo de cifrado, etc. No existe un cmdlet o módulo específico para esta configuración por lo que se realiza mediante el contexto [netsh wlan](#)