

# UNIVERSIDAD AUTÓNOMA DE SINALOA

## INGENIERÍA EN SOFTWARE



**Integrantes:** Gutierrez Valdez José Guadalupe

**Grupo:** 3-01

**Materia:** Administración de sistemas

**Práctica #2:** Automatización y Gestión del Servidor DHCP

**Profesor:** Dr.Herman Geovanny Ayala Zuñiga

**Fecha de Entrega:** 12/02/2026

# Historial de cambios(Github)

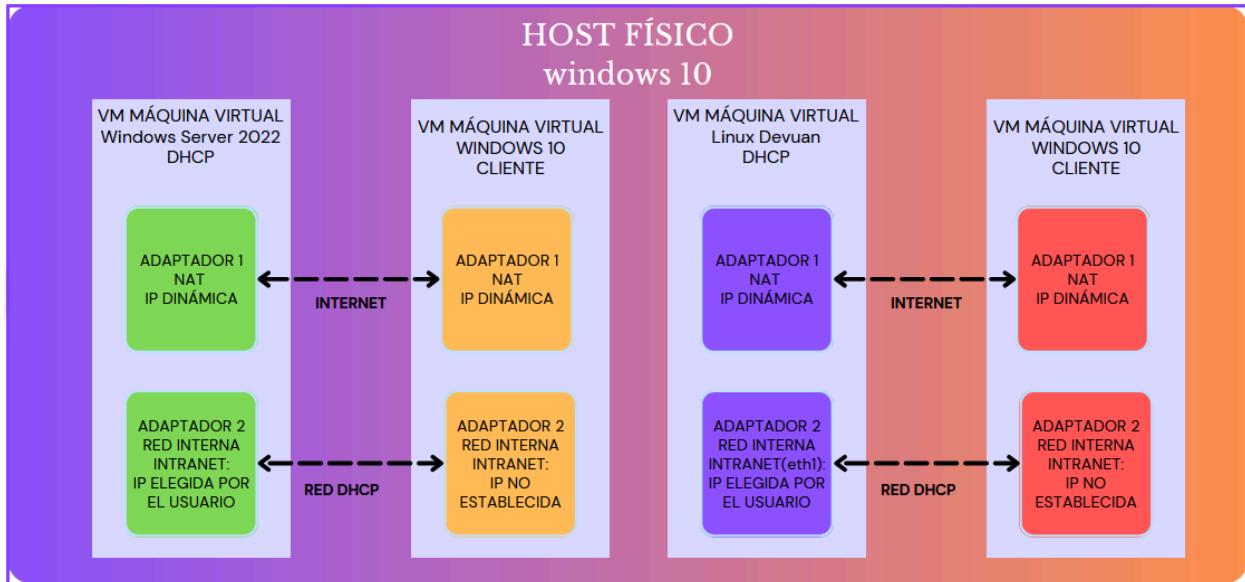
Cambio aplicado	Descripción	Fecha
<i>Creación del repositorio en github</i>	<i>Se creó el repositorio en github con el nombre “entorno-red-sistemas”.</i>	12/02/2026
<i>Carga de los archivos de la práctica #1</i>	<i>Se subieron los archivos al repositorio github, los cuales son capturas, scripts, configuraciones y otro tipo de archivos referente a la práctica #1</i>	12/02/2026

## Objetivo

El objetivo de esta práctica es el crear dos scripts que tendrán la misma función, instalar todo lo necesario para dejar un servidor DHCP configurado y listo para tener comunicación a través de una máquina cliente en la misma red, a la vez de que contendrá las herramientas para monitorear dicha comunicación. Los scripts serán escritos para poder ser ejecutados en bash y powershell respectivamente y deben de permitir una instalación mínima de interacción con el usuario, solo dejando estas interacciones relegadas a las opciones ofrecidas por medio de un menú interactivo.

## Diagrama de topología

En la siguiente imagen se muestra la arquitectura que presentará el sistema de conexión entre los servidores DHCP y la máquina cliente.



## Manual de usuario

### Scripts

#### BASH

[dhcp-funciones.sh](#)

[dhcp-main.sh](#)

requisitos:

- Sistema operativo Linux Devuan instalado
- Tener instalado BASH en sistema operativo
- Ser usuario root
- Tener el script dhcp\_main.sh en la misma carpeta
- Conexión a internet debido a las descargas

Explicación del script:

## VARIABLES

- **#!/bin/bash**
  - Usa Bash para leer este archivo
- **SUBRED=""**
  - Guarda la dirección de red (ej: 192.168.1.0)
- 
- **MASCARA=""**
  - Guarda la máscara de red (ej: 255.255.255.0)
- 
- **GATEWAY=""**
  - Guarda la IP del router/puerta de enlace
- 
- **DNS\_SERVER=""**
  - Guarda la IP del servidor DNS
- 
- **INTERFAZ\_DHCP="eth1"**
  - Guarda el nombre de la tarjeta de red que usará el DHCP
- 
- **NOMBRE\_SCOPE="Red\_Sistemas\_DHCP"**
  - Nombre descriptivo para identificar esta configuración
- 
- **TIEMPO\_CONCESION="7200"**
  - Tiempo en segundos que dura una IP prestada (2 horas)
- 
- **IP\_SERVIDOR=""**
  - Guarda la IP fija del servidor DHCP
- 
- **IP\_RANGO\_INICIO=""**
  - Primera IP del rango que se repartirá
- 
- **IP\_RANGO\_FIN=""**
  - Última IP del rango que se repartirá
- 
- **MAX TIEMPO="14400"**

- Tiempo máximo que una IP puede estar prestada (4 horas)

## FUNCIONES

```
verificar_estado_servicio() {
    local servicio=$1
    Guarda el nombre del servicio que recibió (isc-dhcp-server)

    if /etc/init.d/$servicio status > /dev/null 2>&1; then
        Pregunta al sistema: ¿El servicio está funcionando?
        > /dev/null 2>&1 significa "no me muestres nada, ni errores ni
        resultados"

        return 0
        Responde: "Sí, está activo" (0 = éxito)
    else
        return 1
        Responde: "No, no está activo" (1 = error)
    fi
}

validar_rango_ip() {
    local inicio=$1
    local fin=$2
    Recibe IP de inicio y IP de fin

    if [[ -z "$inicio" || -z "$fin" ]]; then
        Revisa: ¿Falta alguna IP?
        echo "ERROR: IP de inicio o fin no especificada"
        return 1
    fi

    local inicio_num=$(ip_to_num "$inicio")
    local fin_num=$(ip_to_num "$fin")
    Convierte ambas IPs a números para poder compararlas

    if [[ $fin_num -lt $inicio_num ]]; then
        Revisa: ¿La IP final es menor que la inicial?
        echo "ERROR: La IP final ($fin) es menor que la IP inicial ($inicio)"
        return 1
    fi

    return 0
    El rango es válido
}
```

```

}

configurar_interfaz() {
    local interfaz=$1
    local ip=$2
    local mascara=$3
    Recibe: nombre de interfaz, IP, máscara

    echo ""
    echo "==== CONFIGURANDO INTERFAZ DE RED ==="

    if ! ip link show $interfaz > /dev/null 2>&1; then
        ip link show: muestra las interfaces de red
        Pregunta: ¿Existe esta interfaz?

        echo "ERROR: La interfaz $interfaz no existe"
        echo ""
        echo "Interfaces disponibles:"
        ip link show | grep "[0-9].:" | awk '{print $2}' | tr -d ':'
        Muestra todas las interfaces disponibles

        read -p "Ingrese el nombre de la interfaz a usar: " INTERFAZ_DHCP
        Pide al usuario que escriba una interfaz válida

        interfaz=$INTERFAZ_DHCP
        Guarda el nuevo nombre

        if ! ip link show $interfaz > /dev/null 2>&1; then
            echo "ERROR: La interfaz $interfaz tampoco existe"
            return 1
            Verifica si la nueva también es inválida
        fi
    fi

    if ip link show $interfaz | grep -q "state DOWN"; then
        echo "Activando interfaz $interfaz..."
        ip link set $interfaz up
        ¿La interfaz está apagada? La enciende
    fi

    local cidr=24
    Asume máscara 255.255.255.0 = /24

    case $mascara in

```

```

    "255.255.255.255") cidr=32 ;;
    "255.255.255.254") cidr=31 ;;
    ...
esac
Convierte máscara decimal a notación CIDR (/número)

ip addr flush dev $interfaz 2>/dev/null
Borra todas las IPs de la interfaz

ip addr add $ip/$cidr dev $interfaz
Asigna la nueva IP con su CIDR

ip link set $interfaz up
Asegura que la interfaz esté activa

cat > /etc/network/interfaces.d/$interfaz << EOF
Crea archivo de configuración persistente
auto $interfaz
iface $interfaz inet static
    address $ip
    netmask $mascara
EOF
Guarda la configuración para que sobreviva a reinicios

echo "OK Interfaz $interfaz configurada"
return 0
}

instalar_dhcp() {
{
    Agrupa todos los comandos...
} > /dev/null 2>&1
...y redirige TODA la salida a la basura
El usuario no ve NADA de lo que pasa aquí dentro

if dpkg -l | grep -q "isc-dhcp-server"; then
    ¿Ya está instalado?

    if verificar_estado_servicio "isc-dhcp-server"; then
        /etc/init.d/isc-dhcp-server stop > /dev/null 2>&1
        Si está corriendo, lo detiene
    fi
}

```

```
DEBIAN_FRONTEND=noninteractive apt-get remove --purge -y  
isc-dhcp-server > /dev/null 2>&1  
Lo desinstala completamente (sin preguntar)
```

```
DEBIAN_FRONTEND=noninteractive apt-get autoremove -y >  
/dev/null 2>&1  
Limpia paquetes que ya no sirven
```

```
DEBIAN_FRONTEND=noninteractive apt-get autoclean -y >  
/dev/null 2>&1  
Limpia la caché  
fi
```

```
DEBIAN_FRONTEND=noninteractive apt-get update > /dev/null 2>&1  
Actualiza la lista de paquetes (sin preguntar)
```

```
cat > /usr/sbin/policy-rc.d << 'EOF'  
Crea un archivo temporal...  
#!/bin/sh  
exit 101  
EOF  
...que hace que los servicios NO se inicien automáticamente al  
instalarse
```

```
chmod +x /usr/sbin/policy-rc.d  
Lo hace ejecutable
```

```
DEBIAN_FRONTEND=noninteractive apt-get install -y -qq  
isc-dhcp-server > /dev/null 2>&1  
Instala el servidor DHCP  
-y : sí a todo  
-qq : super silencioso
```

```
rm -f /usr/sbin/policy-rc.d  
Borra el archivo temporal
```

```
if [ $? -eq 0 ]; then  
$? contiene el resultado del último comando  
¿La instalación fue exitosa?
```

```
echo "INTERFACESv4=\"$INTERFAZ_DHCP\" >  
/etc/default/isc-dhcp-server  
Configura qué interfaz debe escuchar el DHCP
```

```

update-rc.d -f isc-dhcp-server remove > /dev/null 2>&1
Elimina el inicio automático (lo manejaremos nosotros)

        return 0
    else
        return 1
    fi

    return $?
Devuelve el código de éxito o error
}

```

**Ejecución:** Para ejecutar este script, debemos de ser usuarios root y tener los dos scripts en el misma carpeta luego escribimos

**./dhcp-main.sh en devuan**



Después presionamos enter y aparecerá este menú



Y aparecen varias opciones, si seleccionamos 1, nos dirá si tenemos el dhcp instalado o no

```
==> VERIFICANDO INSTALACION ==>
OK isc-dhcp-server esta instalado
isc-dhcpd-4.4.3-P1
ERROR Servicio DHCP NO esta activo

Presione Enter para continuar...
```

la segunda opción, nos permite descargar todo para ejecutar nuestro servidor de dhcp de manera silenciosa

```
CONFIGURACION DE SERVIDOR DHCP

1. Verificar instalacion del servidor DHCP
2. Instalar servidor DHCP (isc-dhcp-server)
3. Configurar servidor DHCP
4. Monitorear estado del servicio
5. Mostrar configuracion actual
6. Reinstalar servidor DHCP
7. Salir

Seleccione una opcion [1-7]: 2

Presione Enter para continuar....
```

y la tercera opción permite configurar el servidor dhcp y levantar el servidor

```
==> CONFIGURACION DE PARAMETROS DHCP ==>
Nombre del ambito [Red_Sistemas_DHCP]: prueba
Mascara de subred [255.255.255.0]: 255.255.255.0
IP estatica del servidor: 103.5.93.5
Subred calculada: 103.5.93.0
Broadcast calculado: 103.5.93.255
IP de inicio del rango DHCP [103.5.93.6]: 103.5.93.142
IP de fin del rango DHCP [103.5.93.150]: 103.5.93.152
Gateway (puerta de enlace) [103.5.93.1]:
Servidor DNS [8.8.8.8]:
Tiempo de concesion en segundos [7200]: 5000

==> RESUMEN DE CONFIGURACION ==>
Nombre del ambito: prueba
Interfaz: eth1
IP del servidor: 103.5.93.5
Mascara de red: 255.255.255.0
Subred: 103.5.93.0
Broadcast: 103.5.93.255
Rango DHCP: 103.5.93.142 - 103.5.93.152
Gateway: 103.5.93.1
DNS: 8.8.8.8
Tiempo concesion: 5000 segundos

Continuar con la configuracion? (s/n):
```

Con lo que podremos dejar listo el servidor para recibir comunicaciones a través de un cliente en una red local.

## PowerShell

dhcp-main.ps1

requisitos:

- Sistema Windows Server 2022 CLI
- Tener instalado PowerShell 3+ y ejecutado como administrador
- Tener conexión a internet para las descargas

Explicación del script:

- **\$SUBRED = ""**
  - Guarda la dirección de red (ej: 192.168.1.0)
- **\$MASCARA = ""**
  - Guarda la máscara de red (ej: 255.255.255.0)
- **\$GATEWAY = \$null**
  - Guarda la IP del router/puerta de enlace (nulo por defecto)
- **\$DNS\_SERVER = \$null**
  - Guarda la IP del servidor DNS (nulo por defecto)
- **\$INTERFAZ\_DHCP = ""**
  - Guarda el nombre de la tarjeta de red (ej: "Ethernet0")
- **\$NOMBRE\_SCOPE = "Red\_Sistemas\_DHCP"**
  - Nombre descriptivo para identificar el scope DHCP
- **\$TIEMPO\_CONCESION = 7200**
  - Tiempo en segundos que dura una IP prestada (2 horas)
- **\$MAX TIEMPO = 14400**
  - Tiempo máximo que una IP puede estar prestada (4 horas)
- **\$IP\_SERVIDOR = ""**
  - Guarda la IP fija del servidor DHCP
- **\$IP\_RANGO\_INICIO = ""**
  - Primera IP del rango que se repartirá
- **\$IP\_RANGO\_FIN = ""**
  - Última IP del rango que se repartirá
- **\$BROADCAST = ""**
  - Guarda la dirección de broadcast calculada

**function Validar-IP {**

param([string]\$ip)

if ([string]::IsNullOrEmpty(\$ip)) {

Write-Host "ERROR: IP no puede estar vacía"

```

return $false

¿La IP está vacía?

}

$ipPattern = '^([0-9]{1,3})\.( [0-9]{1,3})\.( [0-9]{1,3})\.( [0-9]{1,3})$'

Expresión regular: 4 grupos de 1-3 dígitos separados por puntos

if ($ip -notmatch $ipPattern) {

    Write-Host "ERROR: Formato de IP invalido: $ip"

    return $false

    ¿Tiene formato número.número.número.número?

}

$octets = $ip -split '\.'

Separa en 4 números

foreach ($octet in $octets) {

    $num = [int]$octet

    if ($num -lt 0 -or $num -gt 255) {

        Write-Host "ERROR: Octeto fuera de rango (0-255): $octet"

        return $false

        ¿Cada número está entre 0 y 255?

    }

}

# IPs prohibidas (igual que en Bash)

if ($ip -eq "0.0.0.0") {

    Write-Host "ERROR: IP PROHIBIDA: $ip (0.0.0.0 no se puede usar)"

    return $false
}

```

```

}

if ($octets[0] -eq 127 -and $octets[1] -eq 0 -and $octets[2] -eq 0 -and
[int]$octets[3] -ge 0 -and [int]$octets[3] -le 2) {
    Write-Host "ERROR: IP PROHIBIDA: $ip (Rango de loopback 127.0.0.0-127.0.0.2)"
    return $false
}

if ($ip -eq "255.255.255.255") {
    Write-Host "ERROR: IP PROHIBIDA: $ip (Broadcast global)"
    return $false
}

# Advertencias (no errores)
if ($octets[3] -eq 0) {
    Write-Host "ADVERTENCIA: $ip es una direccion de red"
}

if ($octets[3] -eq 255) {
    Write-Host "ADVERTENCIA: $ip es una direccion de broadcast"
}

return $true

Todo está bien

}

function Calcular-DireccionRed {

    param([string]$ip, [string]$mascara)

    $ipNum = IP-ToUInt32 $ip

```

```

$mascaraNum = IP-ToUInt32 $mascara
$redNum = $ipNum -band $mascaraNum
AND binario: solo deja los bits donde la máscara tiene 1
return (UInt32-ToIP $redNum)

Ejemplo: 192.168.1.10 AND 255.255.255.0 = 192.168.1.0

}

function Verificar-Instalacion {

    Write-Host ""
    Write-Host "==== VERIFICANDO INSTALACION ==="

try {

    $feature = Get-WindowsFeature -Name DHCP -ErrorAction SilentlyContinue
    Get-WindowsFeature: consulta características instaladas en Windows Server
    -Name DHCP: busca el rol DHCP
    -ErrorAction SilentlyContinue: no muestra errores
    $dhcpInstalled = $feature.Installed
    ¿Está instalado? (True/False)

} catch {
    $dhcpInstalled = $false
    Si hay error, asume que no está instalado
}

if ($dhcpInstalled) {
    Write-Host "OK DHCP Server esta instalado"
    $service = Get-Service -Name DHCPServer -ErrorAction SilentlyContinue
}

```

```
Get-Service: consulta servicios del sistema  
-Name DHCPServer: servicio DHCP  
  
  
if ($service -and $service.Status -eq "Running") {  
    Write-Host "OK Servicio DHCP esta activo"  
}  
else {  
    Write-Host "ERROR Servicio DHCP NO esta activo"  
}  
  
try {  
    $version = netsh dhcp server show version | Select-String "Version"  
    netsh: herramienta de configuración de red  
    dhcp server show version: muestra versión del DHCP  
    Select-String: filtra líneas que contienen "Version"  
    Write-Host $version  
}  
catch {}  
return $true  
  
} else {  
    Write-Host "ERROR DHCP Server NO esta instalado"  
    return $false  
}  
}  
  
function Instalar-DHCP {  
    Write-Host ""  
    Write-Host "==== INSTALANDO SERVIDOR DHCP ==="  
}
```

```
if (Verificar-Instalacion) {  
    Write-Host ""  
  
    $respuesta = Read-Host "El servidor DHCP ya esta instalado. Desea reinstalarlo? (s/n)"  
  
    Read-Host: pide entrada al usuario  
  
  
    if ($respuesta -notmatch '^sS$') {  
        Write-Host "Instalacion cancelada"  
  
        return $false  
  
        Si no responde s/S, cancela  
  
    }  
  
  
    Write-Host "Deteniendo servicio..."  
  
    Stop-Service -Name DHCPServer -Force -ErrorAction SilentlyContinue  
  
    Detiene el servicio DHCP a la fuerza  
  
  
    Write-Host "Desinstalando DHCP Server..."  
  
    try {  
        Uninstall-WindowsFeature -Name DHCP -Remove  
  
        Uninstall-WindowsFeature: desinstala características  
  
        -Remove: elimina también archivos  
  
  
        Write-Host "OK DHCP Server desinstalado completamente"  
  
    } catch {
```

```
Write-Host "ERROR: No se pudo desinstalar"  
return $false  
}  
}  
  
Write-Host "Instalando DHCP Server..."  
try {  
    Install-WindowsFeature -Name DHCP -IncludeManagementTools  
    Install-WindowsFeature: instala características  
    -IncludeManagementTools: instala también herramientas de administración  
  
    # Configurar grupos de seguridad  
    try {  
        netsh dhcp add securitygroups | Out-Null  
        netsh dhcp add securitygroups: agrega grupos de seguridad DHCP en AD  
        Out-Null: descarta la salida (no mostrar)  
  
        Write-Host "OK Grupos de seguridad configurados"  
    } catch {}  
  
    Start-Sleep -Seconds 3  
    Espera 3 segundos  
  
    Write-Host "OK DHCP Server instalado correctamente"
```

```

Write-Host ""

Write-Host "NOTA: El servicio no se iniciara automaticamente"

Write-Host "Debe configurar el servidor DHCP para iniciar el servicio"

return $true

} catch {

    Write-Host "ERROR: Fallo la instalacion de DHCP Server"

    Write-Host "Detalles: $_"

    $_ : variable que contiene el último error

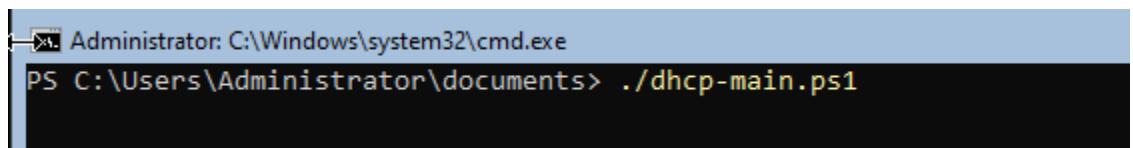
return $false

}
}

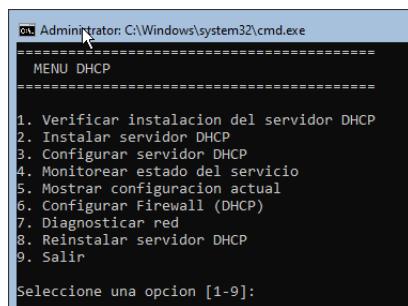
```

### Ejecución:

Para ejecutar este script, ingresamos al PowerShell de nuestro sistema operativo win server 2022 e ingresamos **./dhcp-main.ps1** (recuerda estar en la misma carpeta que el script)



Luego nos aparecerá un menú



y como en script de Devuan, la primera opción nos permite verificar la instalación del dhcp en la máquina

```
Seleccione una opcion [1-9]: 1

==== VERIFICANDO INSTALACION ===
OK DHCP Server esta instalado
OK Servicio DHCP esta activo
The version of the DHCP Server 169.254.0.107 is 10.0.
True

Presione Enter para continuar...:
```

La segunda opción permite instalar los recursos para el DHCP

```
c:\ Administrator: C:\Windows\system32\cmd.exe
Administrator: C:\Windows\system32\cmd.exe
=====
MENU DHCP
=====

1. Verificar instalacion del servidor DHCP
2. Instalar servidor DHCP
3. Configurar servidor DHCP
4. Monitorear estado del servicio
5. Mostrar configuracion actual
6. Configurar Firewall (DHCP)
7. Diagnosticar red
8. Reinstalar servidor DHCP
9. Salir

Seleccione una opcion [1-9]: 2

==== INSTALANDO SERVIDOR DHCP ===

==== VERIFICANDO INSTALACION ===
OK DHCP Server esta instalado
OK Servicio DHCP esta activo
The version of the DHCP Server 169.254.0.107 is 10.0.

El servidor DHCP ya esta instalado. Desea reinstalarlo? (s/n):
```

Y si ya está instalado nos preguntará si queremos volver a instalarlo.

La tercera opción permite configurar el servidor para encenderlo (solo si ya tienes los recursos descargados)

```
Nombre del ambito [Red_Sistemas_DHCP]: prueba
Mascara de subred [255.255.255.0]: 255.255.255.0
ADVERTENCIA: 255.255.255.0 es una direccion de red
IP estatica del servidor DHCP (obligatoria): 103.5.93.5
Subred calculada: 103.5.93.0
Broadcast calculado: 103.5.93.255
IP de inicio del rango DHCP [103.5.93.6]: 103.5.93.142
IP de fin del rango DHCP [103.5.93.150]: 103.5.93.152
Gateway (puerta de enlace) [103.5.93.1]:
OK Sin gateway configurado
Servidor DNS [8.8.8.8]:
OK Sin DNS configurado
Tiempo de concesion en segundos [7200]: 5000

==== RESUMEN DE CONFIGURACION ===
Nombre del ambito: prueba
Interfaz: ethernet 2
IP del servidor: 103.5.93.5
Mascara de red: 255.255.255.0
Subred: 103.5.93.0
Broadcast: 103.5.93.255
Rango DHCP: 103.5.93.142 - 103.5.93.152
Gateway: No configurado
DNS: No configurado
Tiempo concesion: 5000 segundos
Tiempo maximo: 10000 segundos

Continuar con la configuracion? (s/n): .
```

Con lo que podremos dejar listo el servidor para recibir comunicaciones a través de un cliente en una red local.

## Conclusiones

Esta práctica fue algo difícil ya que el crear los scripts y luego ir corrigiendo los errores que aparecían me tomaban bastante tiempo, pero el poder hacer las configuraciones y arreglos correctos me enseñó mucho y pude al final lograr que mis clientes (Devuan y Windows Server) pudieran servir como servidor DHCP, y confirmar su comunicación exitosa a través de un cliente en Windows 10. Esta práctica me hizo utilizar muchas herramientas que no sabía o desconocía su existencia o su poder de alcance.

## Bibliografía

<https://nyaw.wiki.devuan.org/Devuan%20documentation.HTML>

<https://learn.microsoft.com/en-us/powershell/>

<https://www.debian.org/doc/>

[https://es.wikipedia.org/wiki/Direcci%C3%B3n\\_IP](https://es.wikipedia.org/wiki/Direcci%C3%B3n_IP)

[https://es.wikipedia.org/wiki/Protocolo\\_de\\_configuraci%C3%B3n\\_din%C3%A1mica\\_de\\_host](https://es.wikipedia.org/wiki/Protocolo_de_configuraci%C3%B3n_din%C3%A1mica_de_host)