



Universidad Autónoma de Sinaloa

Licenciatura en Ingeniería de Software



Tarea 2: Automatización y Gestión del Servidor DHCP

Materia:

Administración de Sistemas

Nombre del Profesor:

Dr. Herman Geovany Ayala Zúñiga

Nombre del alumno:

Espiritu Manzanarez Diego Manuel

Fecha:

07/02/2026

➤ Introducción y Arquitectura o lo que trate cada práctica

- Historial de Cambios:

Repositorio: <https://github.com/Diegoesp-0/Administracion-de-Sistemas>

Versión	Fecha	Descripción del Cambio	Relación con Git
2.0	07/02/2026	Se crearon las carpetas donde se guardara la tarea 2.	Tarea 2: Carpetas creadas
2.1	08/02/2026	Se guardo en github la beta del script en bash	Tarea 2: Envio de Script beta openSUSE
2.2	08/02/2026	Se guardo en github la segunda versión del script de bash	Tarea 2: Segunda versión del archivo openSUSE
2.3	08/02/2026	Se guardo en github el script de terminado en bash	Tarea 2: Script finalizado openSUSE
2.4	08/02/2026	Se guardaron cosas para crear script en Windows Server	Tarea 2: Subida a Windows server
2.5	08/02/2026	Se mando a github el script ps1 de Windows server terminado	Tarea 2: Subida final de Windows Server
2.6	08/02/2026	Se restructuro el contenido en las carpetas y se guardo	Tarea 2: Estructurado de carpetas
2.7	09/02/2026	Se subió el archivo final de documentación	Tarea 2: Subida de documentación

- **Objetivo:**

Crear un servidor DHCP en el entorno openSUSE y Windows server mediante el uso de scripts en Bash y PowerShell capaz de asignar direcciones IP dinámicas a equipos que se conecten como clientes.

Además de tener opción de monitoreo mostrando el estado del servicio visualizando las conexiones activas para garantizar el correcto funcionamiento del servidor.

- **Diagrama de Topología:**

➤ Guía de Uso de los Scripts

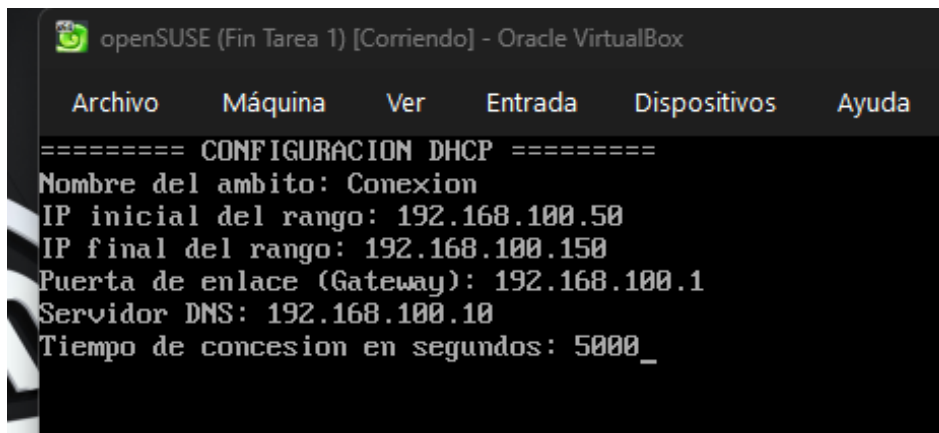
- Requisitos previos:

- Tener conexión a internet
- Saber los parámetros a ingresar
- Saber la tarjeta de red que usaras para el servidor

➤ Instrucciones de Ejecucion:

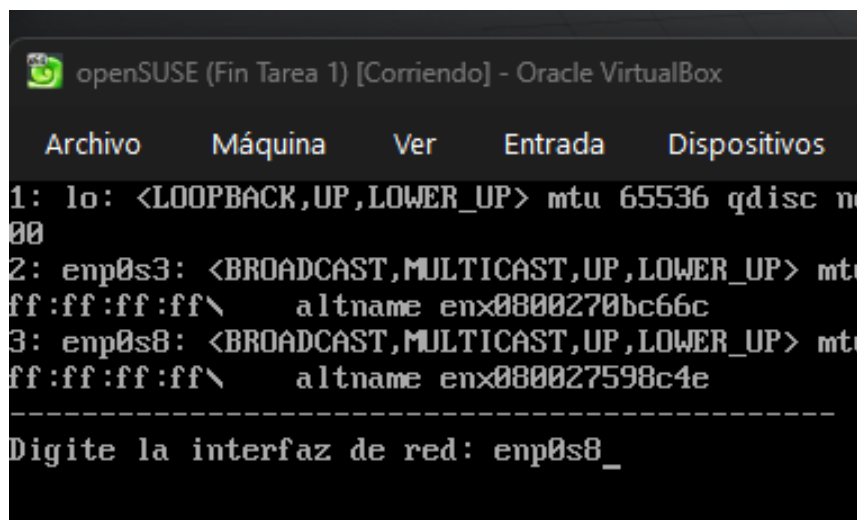
openSUSE:

- 1- Ejecutar el archivo, este evaluara si tienes descargado dhcp-server, en caso de que no, lo descargara, posteriormente ingresar los datos.



```
openSUSE (Fin Tarea 1) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
===== CONFIGURACION DHCP =====
Nombre del ambito: Conexion
IP inicial del rango: 192.168.100.50
IP final del rango: 192.168.100.150
Puerta de enlace (Gateway): 192.168.100.1
Servidor DNS: 192.168.100.10
Tiempo de concesion en segundos: 5000_
```

- 2- Una vez ingresados los datos, deberas de seleccionar la tarjeta de red a usar (en este caso "enp0s8")



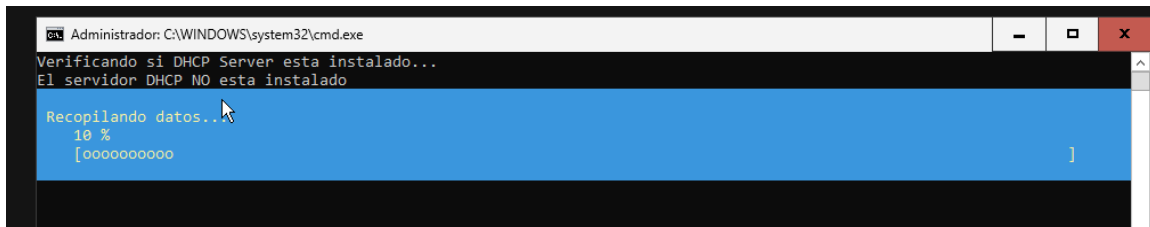
```
openSUSE (Fin Tarea 1) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc n
00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
ff:ff:ff:ff\    altname enx0800270bc66c
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
ff:ff:ff:ff\    altname enx080027598c4e
-----
Digite la interfaz de red: enp0s8_
```

- 3- Una vez ingresados los datos se levantará el servidor DHCP y se mostrará en tiempo real las conexiones y desconexiones.

```
===== MONITOREO =====  
CTRL+C para salir  
starts 1 2026/02/09 04:41:48;  
ends 1 2026/02/09 06:05:08;  
tstp 1 2026/02/09 06:05:08;  
cltt 1 2026/02/09 04:41:48;  
binding state free;  
hardware ethernet 08:00:27:2f:11:ef;  
uid "\001\010\000'\021\357";  
}  
server-duid "\000\001\000\0011\035\011W\010\000'Y\214N";
```

Windows Server:

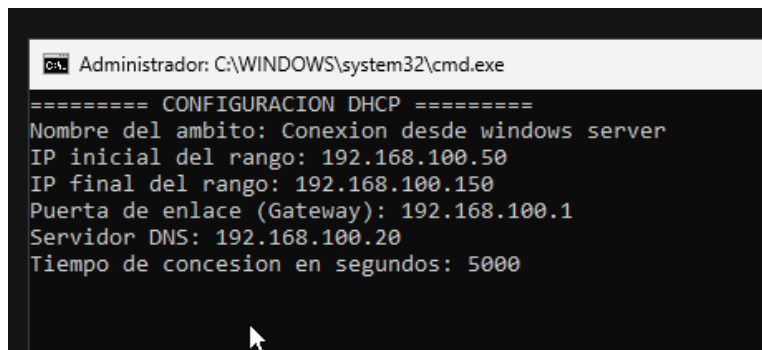
- 1- Ejecutar el archivo .ps1, realizara una comprobación si esta instalado dhcp-server, en caso de que no este, se descargara.



```
Administrador: C:\WINDOWS\system32\cmd.exe
Verificando si DHCP Server esta instalado...
El servidor DHCP NO esta instalado

Recopilando datos...
10 %
[oooooooooooo]
```

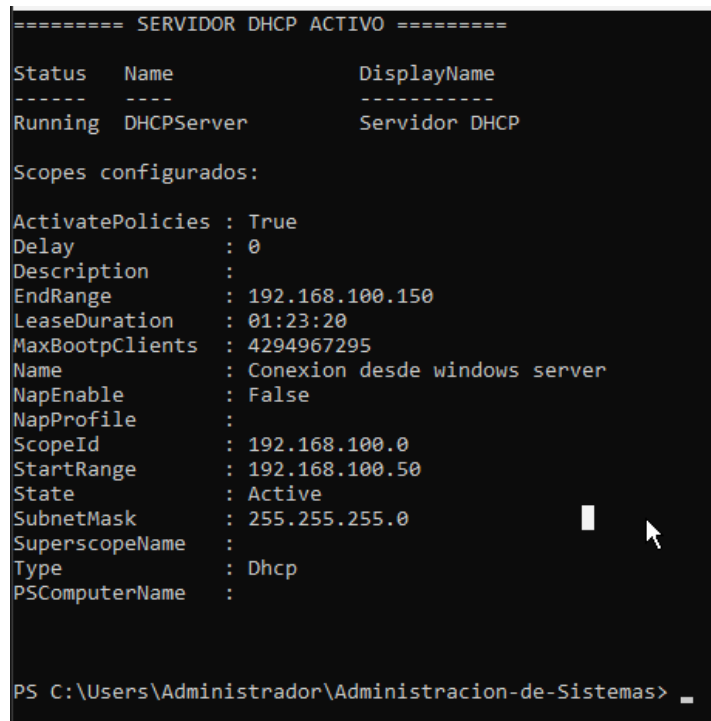
- 2- Ingresar los datos de configuración.



```
Administrador: C:\WINDOWS\system32\cmd.exe

===== CONFIGURACION DHCP =====
Nombre del ambito: Conexion desde windows server
IP inicial del rango: 192.168.100.50
IP final del rango: 192.168.100.150
Puerta de enlace (Gateway): 192.168.100.1
Servidor DNS: 192.168.100.20
Tiempo de concesion en segundos: 5000
```

- 3- Ya estaría activado el servidor



```
===== SERVIDOR DHCP ACTIVO =====

Status      Name           DisplayName
-----
Running     DHCPServer     Servidor DHCP

Scopes configurados:

ActivatePolicies : True
Delay             : 0
Description       :
EndRange          : 192.168.100.150
LeaseDuration     : 01:23:20
MaxBootpClients   : 4294967295
Name              : Conexion desde windows server
NapEnable         : False
NapProfile        :
ScopeId           : 192.168.100.0
StartRange        : 192.168.100.50
State             : Active
SubnetMask        : 255.255.255.0
SuperscopeName    :
Type              : Dhcp
PSComputerName    :

PS C:\Users\Administrador\Administracion-de-Sistemas>
```

➤ Bitácora de Desarrollo y Configuración:

- Explicación del Script:

openSUSE:

```
if zypper se -i dhcp-server | grep -q dhcp-server
then
    echo ""
    echo "El servidor ya esta instalado :D"
else
    echo ""
    echo "El servidor no esta instalado"
    echo "Instalando..."
    sudo zypper install dhcp-server
fi
```

El primer paso es comprobar si esta instalado dhcp-server, esto se hace con esta función, se hace una búsqueda discreta y en caso de que este instalada muestra un mensaje de éxito, si no, pone un mensaje de error y comienza a instalarlo.

```
validar_ip(){
local ip=$1
local verificar=1

if [[ $ip =~ ^([0-9]{1,3}\.){3}[0-9]{1,3}$ ]];
then

    # Dividir la ip en partes
    IFS='.' read -r p1 p2 p3 p4 <<< "$ip"

    # Que los numeros esten maximo hasta 255
    if [[ $p1 -le 255 && $p2 -le 255 && $p3 -le 255 && $p4 -le 255 ]]
    then
        verificar=0
    fi
fi

return $verificar
}
```

Después esta esta función la cual valida que las ips ingresadas sean validas, esto con ayuda del regex, primero se comprueba el patrón, una vez que el patrón es valido pasa a una segunda evaluación donde se separa cada dato separado por el punto y se evalúa si son menores a 255, en caso de que sea correcto, se confirmara como una ip valida.

```

while true
do
    read -p "IP inicial del rango: " IPinicial
    if validar_ip $IPinicial
    then
        break
    else
        echo "IP no valida"
        sleep 1
    fi
done

```

Este bloque de código captura la ip ingresada, en este caso la ip inicial del rango, y la manda a la función validar_ip() para su evaluación, si la ip es correcta avanza el programa, en caso de que no sea correcta se envía un mensaje y se activa el “while” hasta ingresar una ip valida.

Este mismo bloque de código es igual para la captura de ip final del rango, DNS y el Gateway.

```

red="${IPinicial%.*}.0"
mascara="255.255.255.0"

```

```

sudo bash -c "cat > /etc/dhcpd.conf" << EOF
authoritative;

```

```

default-lease-time $LEASE;
max-lease-time $LEASE;

```

```

subnet $red netmask $mascara {
    range $IPinicial $IPfinal;
    option routers $GATEWAY;
    option domain-name-servers $DNS;
}
EOF

```

Ya capturados los datos iniciamos la configuración del servidor, primero se declara una variable red la cual contendrá los primeros 3 campos de la ip siendo el cuarto campo un 0, esto se logra extrayendo primero los 3 campos de la ip y concatenando un “.0”, además se crea una variable de mascara que será de 255.255.255.0 ya que la ip será /24.

Después se edita el archivo de configuración del servidor dhcp con los datos capturados entrando desde su ruta de origen.


```
ip -o link show
```

```
echo "-----"  
read -p "Digite la interfaz de red: " INTERFAZ
```

```
sudo sed -i "s/^DHCPD_INTERFACE=.* /DHCPD_INTERFACE=\"$INTERFAZ\"/"  
/etc/sysconfig/dhcpd
```

```
sudo systemctl enable dhcpd  
sudo systemctl restart dhcpd  
clear
```

```
echo ""  
echo "===== MONITOREO ====="  
echo "CTRL+C para salir"
```

```
tail -f /var/lib/dhcp/db/dhcpd.leases
```

Como último paso, en esta parte se muestran las tarjetas de red disponibles y se le pide al usuario escoger una, ya que se captura el dato se sobrescribe el archivo dhcpd poniéndole la tarjeta seleccionada quedando algo así:

HCPD_INTERFACE= enp0s8

Una vez ingresada la tarjeta se procede a habilitar el servidor dhcpd y reiniciarlo, después empieza el análisis en tiempo real de clientes conectados y desconectados mostrando sus datos.

Windows Server:

```
Clear-Host
Write-Host "Verificando si DHCP Server esta instalado..."

$dhcp = Get-WindowsFeature -Name DHCP

if ($dhcp.Installed) {
    Write-Host "El servidor DHCP ya esta instalado :D"
} else {
    Write-Host "El servidor DHCP NO esta instalado"
    Write-Host "Instalando..."
    Install-WindowsFeature -Name DHCP -IncludeManagementTools
}
```

Primero se comprueba que este instalado el paquete dhcp, en caso de que no este descargado se procederá a descargar automáticamente desde el ManagementTools.

```
function Validar-IP {
    param ([string]$ip)

    if ($ip -match '^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$') {
        $partes = $ip.Split('.')
        foreach ($p in $partes) {
            if ([int]$p -gt 255) {
                return $false
            }
        }
        return $true
    }
    return $false
}
```

Esta parte al igual que con bash utiliza regex para evaluar las ips ingresadas, primero se evalúa la estructura de la ip, para después separar con ayuda del Split cada segmento de la ip para verificar que su numero sea menor a 255, en caso de que todo este correcto continuara el programa.

```
do {  
    $IPinicial = Read-Host "IP inicial del rango"  
} while (-not (Validar-IP $IPinicial))
```

```
do {  
    $IPfinal = Read-Host "IP final del rango"  
} while (-not (Validar-IP $IPfinal))
```

```
do {  
    $GATEWAY = Read-Host "Puerta de enlace (Gateway)"  
} while (-not (Validar-IP $GATEWAY))
```

```
do {  
    $DNS = Read-Host "Servidor DNS"  
} while (-not (Validar-IP $DNS))
```

Esta parte es en la captura de parámetros, pide una ip para cada parámetro de configuración y la manda a la función Validar-IP, en caso de que todo este correcto continuara el programa, en caso de que exista un problema, se activara el do-while para preguntar nuevamente.

```
$red = ($IPinicial.Split('.')[0..2] -join '.') + ".0"  
$mascara = "255.255.255.0"
```

```
if (-not (Get-DhcpServerv4Scope -Scopeld $red -ErrorAction SilentlyContinue)) {  
    Add-DhcpServerv4Scope `  
        -Name $SCOPE `  
        -StartRange $IPinicial `  
        -EndRange $IPfinal `  
        -SubnetMask $mascara `  
        -State Active  
}
```

Aquí se genera la variable mascara como 255.255.255.0 porque la configuración será con /24, y la variable 3 que realizara un Split para concatenar los 3 primeros campos de la ip, en este caso "192.168.100" para después concatenarle un ".0"

Después se realiza un condicional el cual verifica si el ámbito dhcp ya esta creado, y en caso de que no exista lo crea, esto con el objetivo de evitar errores por duplicaciones, su creación se realiza poniendo los datos anteriormente capturados en los campos correspondientes.

```
Set-DhcpServerv4OptionValue `
  -ScopeId $red `
  -Router $GATEWAY `
  -DnsServer $DNS
```

```
Set-DhcpServerv4Scope `
  -ScopeId $red `
  -LeaseDuration ([TimeSpan]::FromSeconds($LEASE))
```

```
Restart-Service DHCPService
```

```
Clear-Host
Write-Host "===== SERVIDOR DHCP ACTIVO ====="
Get-Service DHCPService
```

```
Write-Host ""
Write-Host "Scopes configurados:"
Get-DhcpServerv4Scope
```

La primera parte de este bloque se refiere a las opciones dhcp que recibirán los clientes junto con la ip, la siguiente parte "Set-DhcpServerv4Scope "` Asigna el tiempo de lease de cada cliente, esto convirtiendo los segundos en un formato TimeSpan. Después reiniciamos el servidor para procedentemente mostrar el estado del servicio y mostrar los ámbitos dhcp creados.

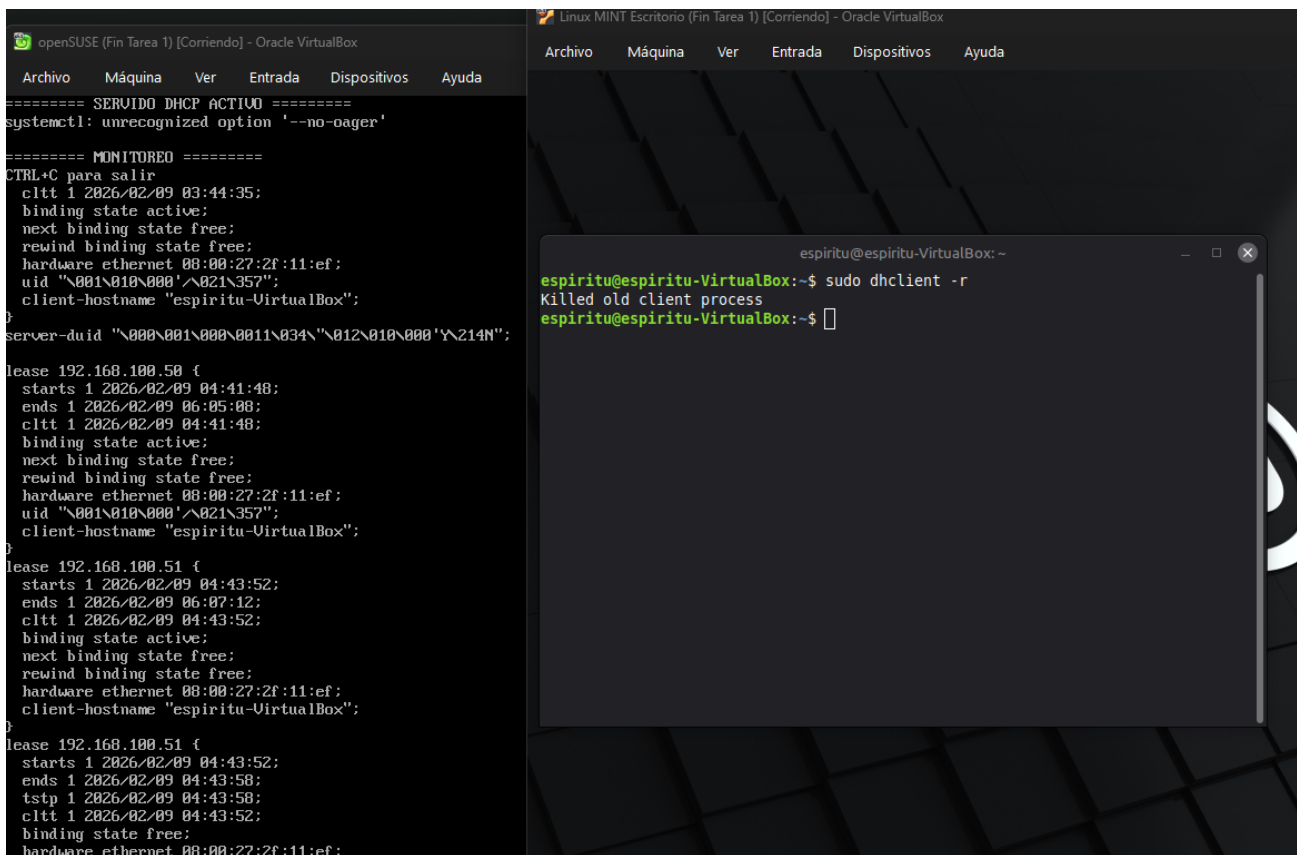
- Evidencias de Configuración:

OpenSUSE:

1- Comprobación de instalación de dhcp-server

```
Despiritu@Srv-Linux-Sistemas:~> rpm -qa | grep dhcp
dhcp-server-4.4.2.P1-160000.2.2.x86_64
dhcp-4.4.2.P1-160000.2.2.x86_64
Despiritu@Srv-Linux-Sistemas:~> _
```

2- Prueba de conexión y desconexión de cliente (Linux Mint)



```
openSUSE (Fin Tarea 1) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
===== SERVIDO DHCP ACTIVO =====
systemctl: unrecognized option '--no-oager'
===== MONITOREO =====
CTRL+C para salir
cltt 1 2026/02/09 03:44:35;
binding state active;
next binding state free;
rewind binding state free;
hardware ethernet 08:00:27:2f:11:ef;
uid "\001\010\000'\^021\357";
client-hostname "espiritu-VirtualBox";
server-duid "\000\001\000\0011\034\^012\010\000'\^214N";
lease 192.168.100.50 {
  starts 1 2026/02/09 04:41:48;
  ends 1 2026/02/09 06:05:08;
  cltt 1 2026/02/09 04:41:48;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 08:00:27:2f:11:ef;
  uid "\001\010\000'\^021\357";
  client-hostname "espiritu-VirtualBox";
}
lease 192.168.100.51 {
  starts 1 2026/02/09 04:43:52;
  ends 1 2026/02/09 06:07:12;
  cltt 1 2026/02/09 04:43:52;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 08:00:27:2f:11:ef;
  client-hostname "espiritu-VirtualBox";
}
lease 192.168.100.51 {
  starts 1 2026/02/09 04:43:52;
  ends 1 2026/02/09 04:43:58;
  tstp 1 2026/02/09 04:43:58;
  cltt 1 2026/02/09 04:43:52;
  binding state free;
  hardware ethernet 08:00:27:2f:11:ef;
}

Linux MINT Escritorio (Fin Tarea 1) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
espiritu@espiritu-VirtualBox: ~
espiritu@espiritu-VirtualBox:~$ sudo dhclient -r
Killed old client process
espiritu@espiritu-VirtualBox:~$
```

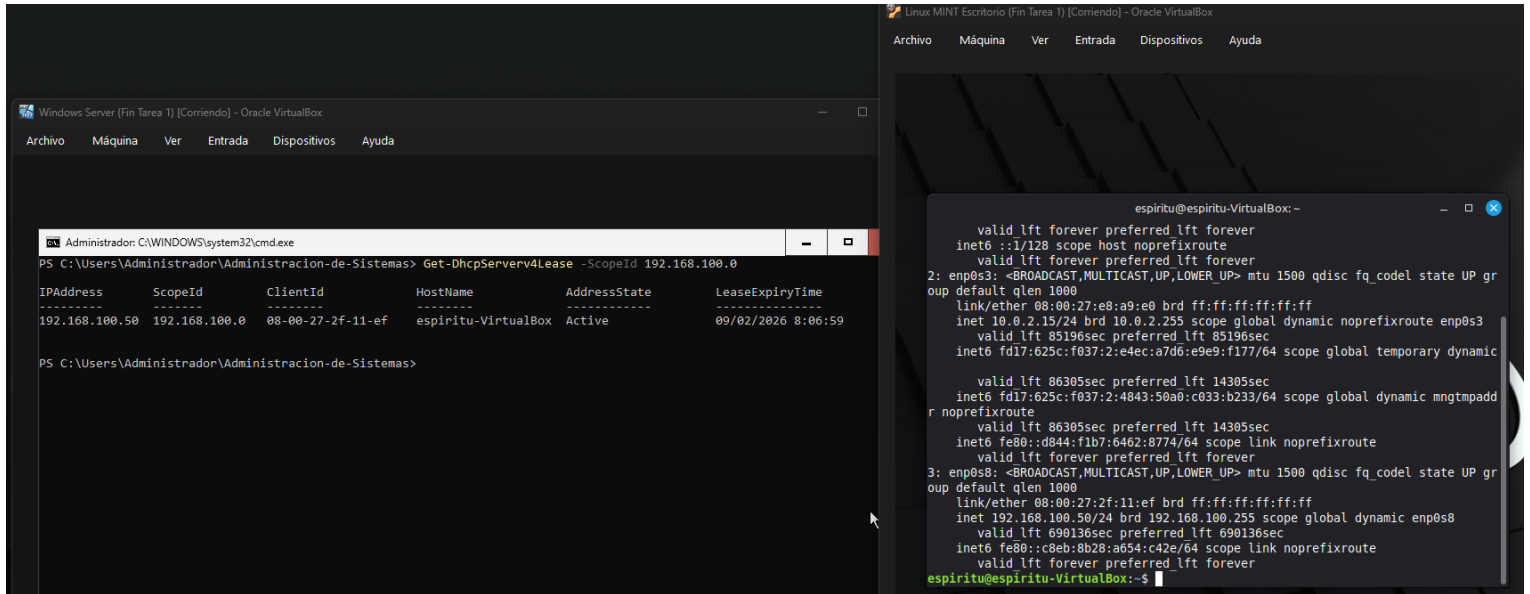
3- Asignación manual de tarjeta de red:

```
GNU nano 8.6 /etc
## Path:      Network/DHCP/DHCP server
## Description: DHCPv4 server settings
## Type:      string
## Default:   ""
## ServiceRestart: dhcpd
#
# Interface(s) for the DHCPv4 server to listen on.
#
# A special keyword is ANY, it will cause dhcpd to autodetect available
# interfaces.
#
# Examples: DHCPD_INTERFACE="eth0 eth1 eth2"
#           DHCPD_INTERFACE="ANY"
#
DHCPD_INTERFACE="enp0s8"

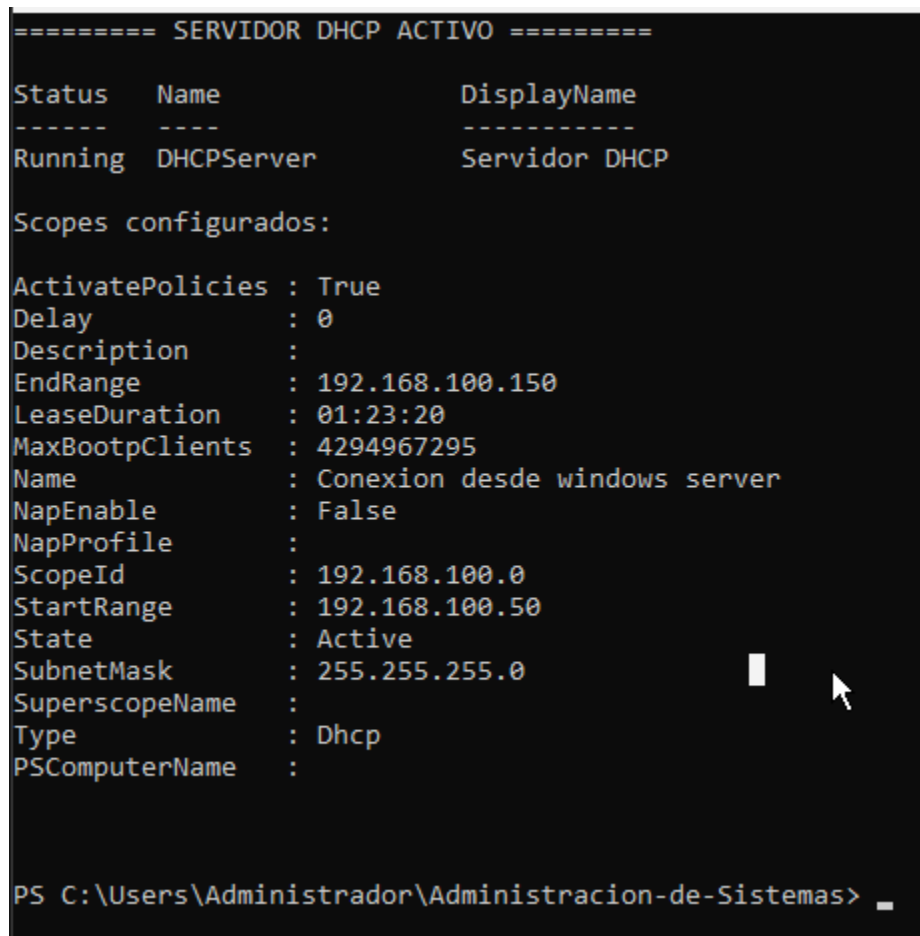
## Path:      Network/DHCP/DHCP server
## Description: DHCPv6 server settings
## Type:      string
## Default:   ""
## ServiceRestart: dhcpd6
#
# Interface(s) for the DHCPv6 server to listen on.
#
# A special keyword is ANY, it will cause dhcpd to autodetect available
# interfaces.
#
```

Windows Server:

1- Prueba de conexión de cliente



2- Servidor activado



➤ Conclusiones y Referencias

- Lecciones Aprendidas:

Tuve como dificultad que la ruta de la configuración ya que se supone que estaría en /etc/dhcp/dhcpd.conf y en realidad yo la tenía en /etc/dhcpd.conf por lo que estuve mucho tiempo buscando porque me daba error sin darme cuenta que solo era eso, además ahora tengo la idea de como es la estructura de la configuración de un servidor dhcp.

- Bibliografía:

<https://linuxconfig.org/what-is-dhcp-and-how-to-configure-dhcp-server-in-linux>

<https://kodekloud.com/blog/regex-shell-script/>

<https://learn.microsoft.com/es-es/windows-server/networking/technologies/dhcp/quickstart-install-configure-dhcp-server?tabs=powershell>

<https://www.w3schools.com/bash/>