



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>

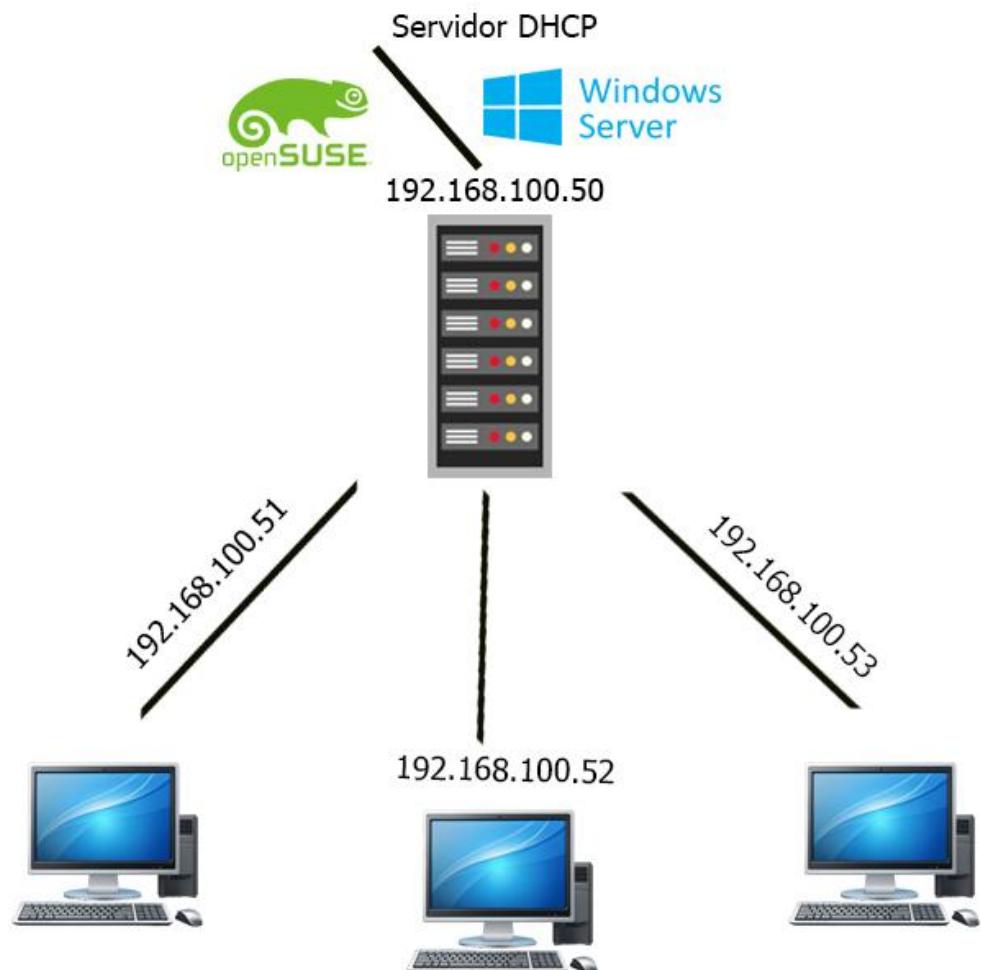
Versión	Fecha	Descripción del Cambio	Relación con Git
2.0	07/02/2026	Se crearon las carpetas donde se guardara la tarea 2.	Tarea 2: Carpetas creadas
2.1	08/02/2026	Se guardo en github la beta del script en bash	Tarea 2: Envio de Script beta openSUSE
2.2	08/02/2026	Se guardo en github la segunda versión 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
2.8	12/02/2026	Se corrigieron los archivos y se organizaron los archivos en las carpetas	Tarea 2: Reestructuracion

- **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
- Conocer las rutas de los archivos de configuracion

➤ Instrucciones de Ejecucion:

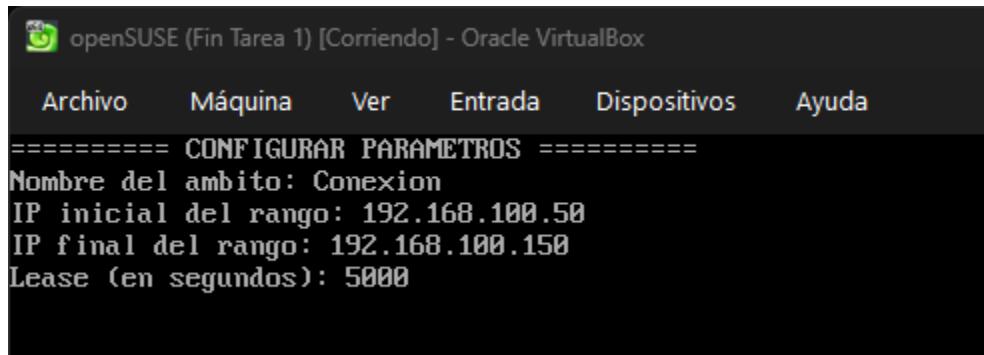
openSUSE:

- 1- Ejecutar el archivo con el comando “verificar” (bash DHCP.sh verificar) para verificar si esta instalado el servicio.



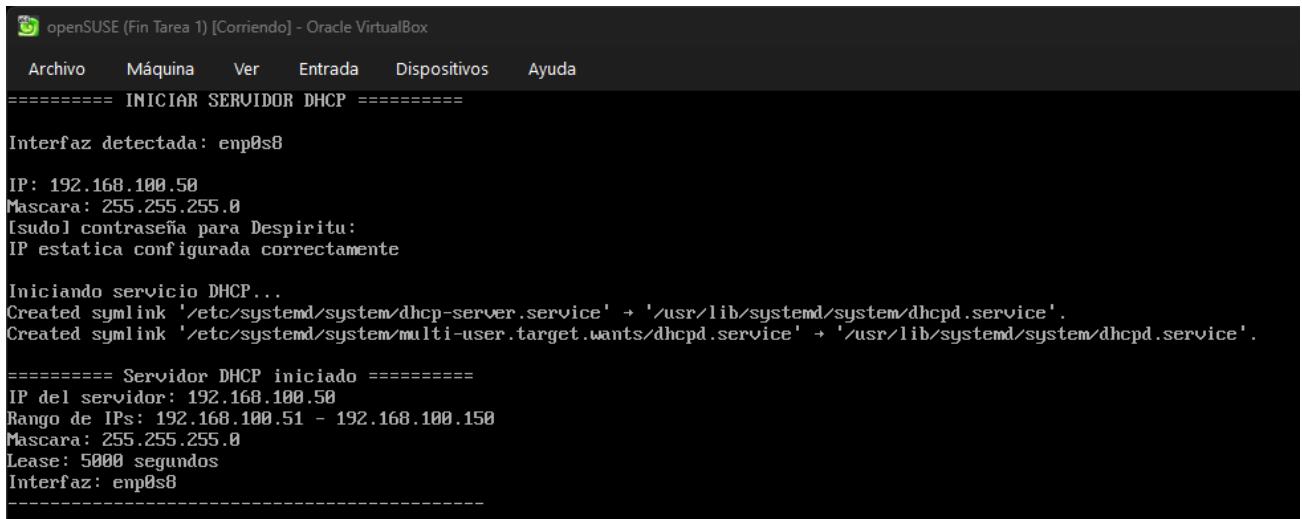
```
openSUSE (Fin Tarea 1) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
El paquete DHCP-SERVER no esta instalado
Desea descargar DHCP-SERVER? (S/s): _
```

- 2- Ejecutar el comando “bash DHCP.sh parametrosconf” para ingresar los parámetros del servidor dhcp



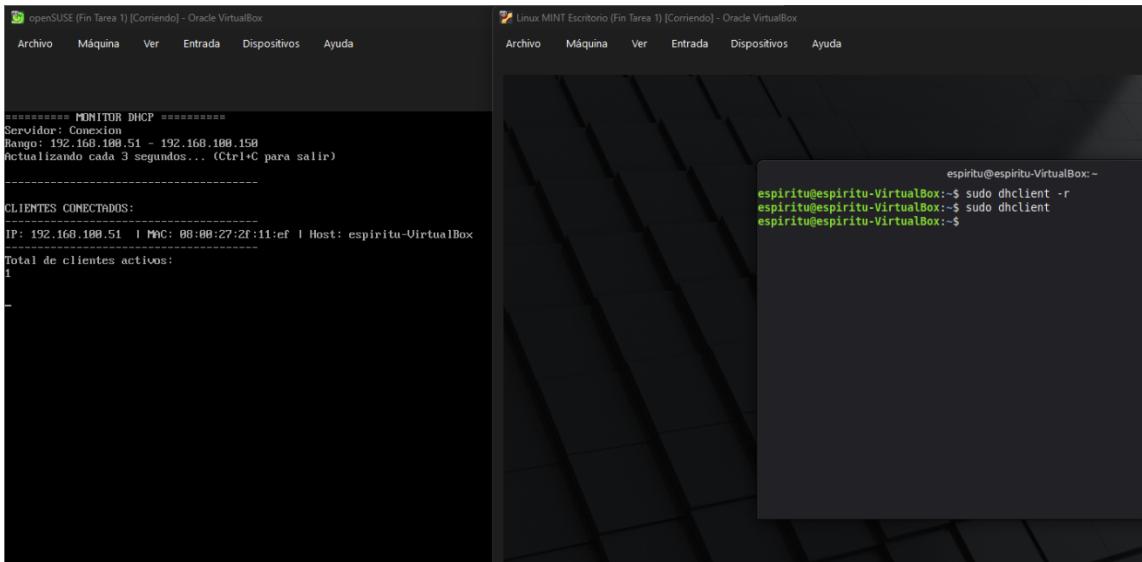
```
openSUSE (Fin Tarea 1) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
=========
CONFIGURAR PARAMETROS ======
Nombre del ambito: Conexion
IP inicial del rango: 192.168.100.50
IP final del rango: 192.168.100.150
Lease (en segundos): 5000
```

3- Usar el comando “bash DHCP.sh iniciar” para comenzar el servicio dhcp.



```
openSUSE (Fin Tarea 1) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
===== INICIAR SERVIDOR DHCP =====
Interfaz detectada: enp0s8
IP: 192.168.100.50
Máscara: 255.255.255.0
[usuario] contraseña para Despiritu:
IP estatica configurada correctamente
Iniciando servicio DHCP...
Created symlink '/etc/systemd/system/dhcp-server.service' → '/usr/lib/systemd/system/dhcpd.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/dhcpd.service' → '/usr/lib/systemd/system/dhcpd.service'.
===== Servidor DHCP iniciado =====
IP del servidor: 192.168.100.50
Rango de IPs: 192.168.100.51 - 192.168.100.150
Máscara: 255.255.255.0
Lease: 5000 segundos
Interfaz: enp0s8
```

4- Ejecutar el comando “bash DHCP.sh monitor” para monitorear las personas que se conecten al servidor dhcp.

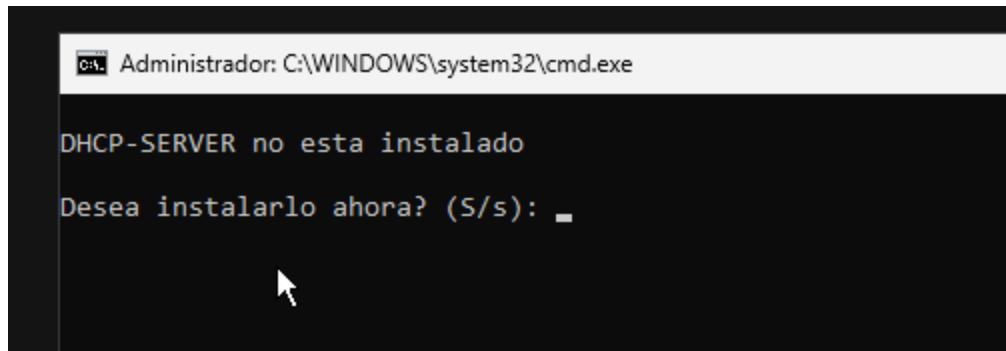


```
openSUSE (Fin Tarea 1) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
===== MONITOR DHCP =====
Servidor: Conexion
Rango: 192.168.100.51 - 192.168.100.150
Actualizando cada 3 segundos... (Ctrl+C para salir)
-----
CLIENTES CONECTADOS:
IP: 192.168.100.51 | MAC: 00:0C:27:2F:11:EF | Host: espiritu-VirtualBox
Total de clientes activos:
1

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

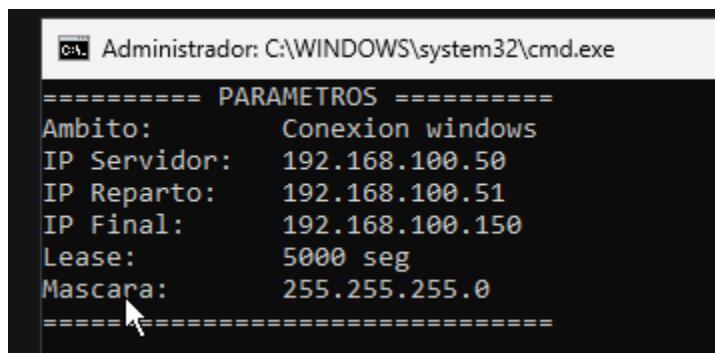
Windows Server:

- 1- Ejecutar el archivo DHCP-WINDOWS.ps1 y seleccionar la opción 1 para verificar que este instalado el servicio.



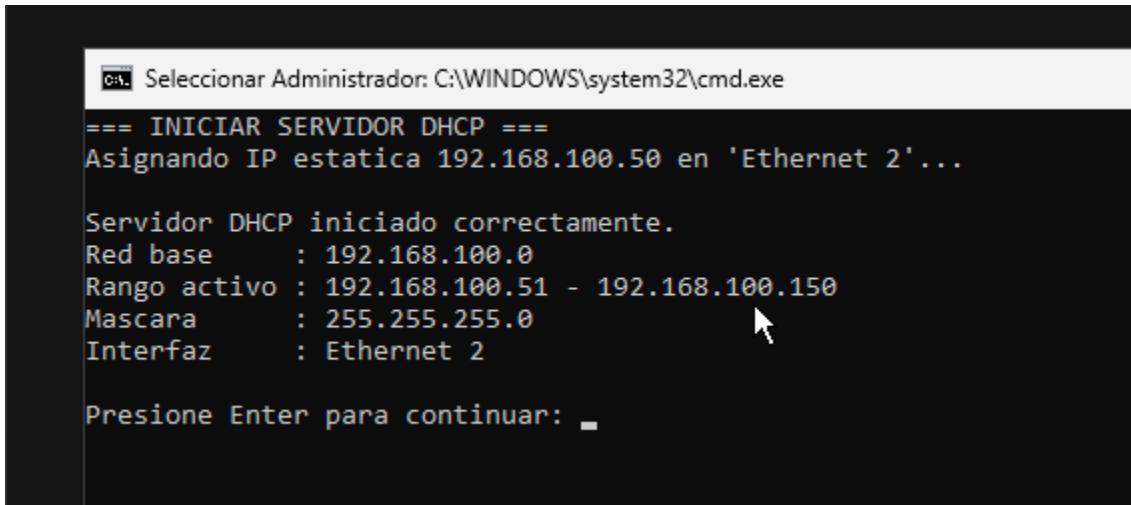
```
DHCP-SERVER no esta instalado
Desea instalarlo ahora? (S/s):
```

- 2- Ditar la opción 3 “Configurar parámetros” e ingresar los parámetros deseados.



```
===== PARAMETROS =====
Ambito: Conexion windows
IP Servidor: 192.168.100.50
IP Reparto: 192.168.100.51
IP Final: 192.168.100.150
Lease: 5000 seg
Mascara: 255.255.255.0
=====
```

3- Seleccionar la opción 4 “Iniciar servidor DHCP”

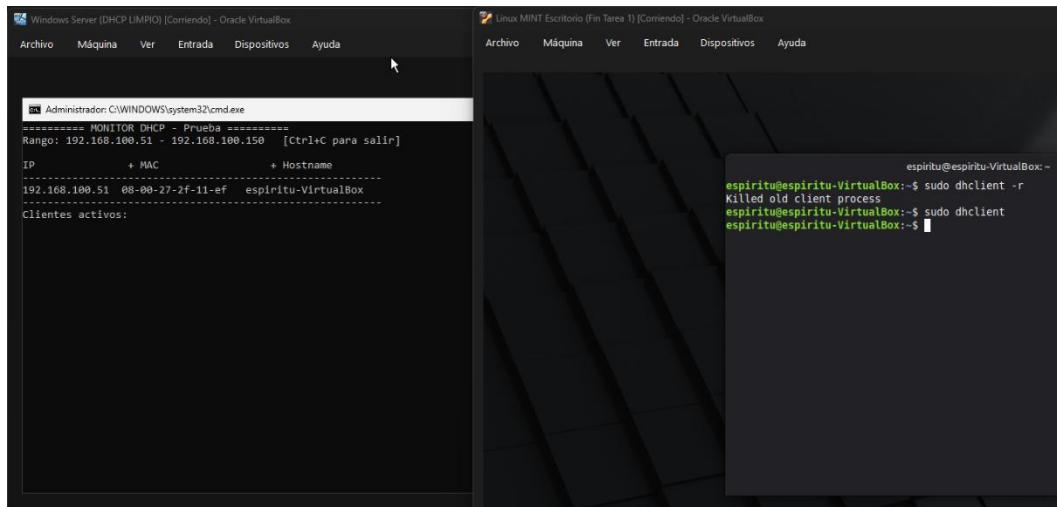


```
Selezionare Administrador: C:\WINDOWS\system32\cmd.exe
== INICIAR SERVIDOR DHCP ==
Asignando IP estatica 192.168.100.50 en 'Ethernet 2'...

Servidor DHCP iniciado correctamente.
Red base      : 192.168.100.0
Rango activo  : 192.168.100.51 - 192.168.100.150
Mascara       : 255.255.255.0
Interfaz      : Ethernet 2

Presione Enter para continuar: -
```

4- Seleccionar la opción 6 “Monitor de clientes”, ahí se reflejarán los clientes conectados y desconectados



➤ Bitácora de Desarrollo y Configuración:

- Explicación del Script (partes relevantes):

openSUSE:

```
verificar(){
    clear
    if zypper se -i dhcp-server > /dev/null 2>&1; then
        echo ""
        echo "DHCP-SERVER esta instalado :D"
        echo ""
    else
        echo ""
        echo "El paquete DHCP-SERVER no esta instalado"
        echo ""
        read -p "Desea descargar DHCP-SERVER? (S/s): " OPC
        if [ "$OPC" = "S" ] || [ "$OPC" = "s" ]; then
            echo "Descargando..."
            sudo zypper install -y dhcp-server
        fi
    fi
}
```

El primer paso es comprobar si está instalado dhcp-server, esto se hace con esta función, se hace una búsqueda 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

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

    IFS='.' read -r p1 p2 p3 p4 <<< "$ip"

    for p in $p1 $p2 $p3 $p4; do
        [[ $p -le 255 ]] || return 1
    done

    [[ $p1 -eq 0 && $p2 -eq 0 && $p3 -eq 0 && $p4 -eq 0 ]] && return 1
    [[ $p1 -eq 255 && $p2 -eq 255 && $p3 -eq 255 && $p4 -eq 255 ]] && return 1
    [[ $p1 -eq 127 ]] && return 1

    return 0
}

```

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 y si usa la ip local 127, en caso de que sea correcto, se confirmara como una ip valida.

```

validar_rango(){
    local ip_inicio=$1
    local ip_fin=$2

    validar_ip "$ip_inicio" || return 1
    validar_ip "$ip_fin" || return 1

    local n_inicio
    local n_fin

    n_inicio=$((ip_a_numero "$ip_inicio"))
    n_fin=$((ip_a_numero "$ip_fin"))

    [[ $n_inicio -lt $n_fin ]]
}

```

Esta función valida que el rango de direcciones sea correcto, primero verifica que ambas IP sean validas, luego las convierte a formato numerico entero para poder compararlas matematicamente. Finalmente comprueba que la IP inicial sea menor que la IP final.

```

sudo tee /etc/dhcpd.conf > /dev/null << EOF
ddns-update-style none;
authoritative;
default-lease-time $LEASE;
max-lease-time $LEASE;

subnet $red netmask $MASCARA {
    range $ip_reparto $IPFINAL;
    option subnet-mask $MASCARA;
    option broadcast-address $broadcast;
}
EOF

```

En esta parte se escriben los parámetros para configurar el servidor sobrescribiendo directamente en el archivo de configuración de dhcp-server (esto dentro de la función iniciar_servidor).

```

if systemctl is-active --quiet dhcpd.service; then
    sudo systemctl enable dhcpd.service
    echo ""
    echo "===== Servidor DHCP iniciado ====="
    echo "IP del servidor: $IPINICIAL"
    echo "Rango de IPs: $ip_reparto - $IPFINAL"
    echo "Mascara: $MASCARA"
    [[ "$GATEWAY" != "X" ]] && echo "Gateway: $GATEWAY"
    [[ "$DNS" != "X" ]] && echo "DNS: $DNS"
    [[ "$DNS2" != "X" ]] && echo "DNS 2: $DNS2"
    echo "Lease: $LEASE segundos"
    echo "Interfaz: $interfaz"
    echo "-----"
    echo ""
else
    echo ""
    echo "ERROR: No se pudo iniciar el servidor DHCP"
    echo ""
    return 1
fi

```

Esto verifica si se inicio el servidor, dentro del caso de éxito hay validaciones para el Gateway y los dns, ya que en caso de que se omita su asignación no se mostraran en la pantalla.

```
local lease_file="/var/lib/dhcp/db/dhcpd.leases"
```

Dentro de la función monitor se encuentra esa línea, la cual consulta los clientes conectados (leases) directamente en el archivo dhcpd.leases, esto lo hace cada 3 segundos en un while.

```
if (ip != "" && mac != "") {  
    activo = 0  
    if (ends == "never") {  
        activo = 1  
    } else if (ends != "") {  
        timestamp_fin = fecha_a_timestamp(ends)  
        if (timestamp_fin > ahora) {  
            activo = 1  
        }  
    }  
    if (activo == 1) {  
        printf "IP: %-15s | MAC: %-17s | Host: %s\n", ip, mac, (hostname != "" ?  
hostname : "Desconocido")  
        clientes_activos++  
    }  
    ip = ""  
    mac = ""  
    hostname = ""  
    ends = ""  
}
```

Este bloque verifica si un cliente DHCP sigue activo antes de mostrarlo en el monitor. Primero comprueba que existan datos válidos de IP y MAC. Luego asume que el cliente no está activo (activo = 0) y evalúa si la concesión no expira (ends == "never") o si su fecha de expiración aún no ha pasado comparándola con la hora actual. Si la concesión sigue vigente, imprime la IP, la MAC y el hostname y aumenta el contador de clientes activos.

Windows Server:

```
function DHCP-Instalado {  
    $feature = Get-WindowsFeature -Name DHCP -ErrorAction SilentlyContinue  
    if ($feature -and $feature.Installed) {  
        return $true  
    } else {  
        return $false  
    }  
}
```

Este bloque define una función que verifica si el rol DHCP Server está instalado en el sistema. Utiliza Get-WindowsFeature para consultar el estado del rol y devuelve verdadero o falso según corresponda.

```
New-NetIPAddress `  
-InterfaceAlias $INTERFAZ `  
-IPAddress $IPINICIAL `  
-PrefixLength (Calcular-CIDR $MASCARA) `  
-DefaultGateway $GATEWAY
```

Este bloque asigna una dirección IP fija al servidor en la interfaz de red seleccionada ya que así aseguramos que sea una ip valida según los parámetros que se ingresaron en un principio.

```
Add-DhcpServerv4Scope `  
-Name $SCOPE `  
-StartRange $IPINICIAL `  
-EndRange $IPFINAL `  
-SubnetMask $MASCARA `  
-LeaseDuration ([TimeSpan]::FromSeconds([int]$LEASE)) `  
-State Active
```

Esta parte es la que activa el servidor dhcp dándole los parámetros que ingreso el usuario en un principio.

```
$leases = Get-DhcpServerv4Lease -Scopelid $SCOPE |  
    Where-Object { $_.AddressState -eq "Active" }  
  
foreach ($lease in $leases) {  
    Write-Host "IP:" $lease.IPAddress `| MAC:" $lease.ClientId `| Host:" $lease.HostName  
}
```

Este bloque permite monitorear las concesiones activas del servidor y muestra la dirección ip asignada, la mac del cliente, y el nombre del equipo, esta verificación se hace cada 3 segundos.

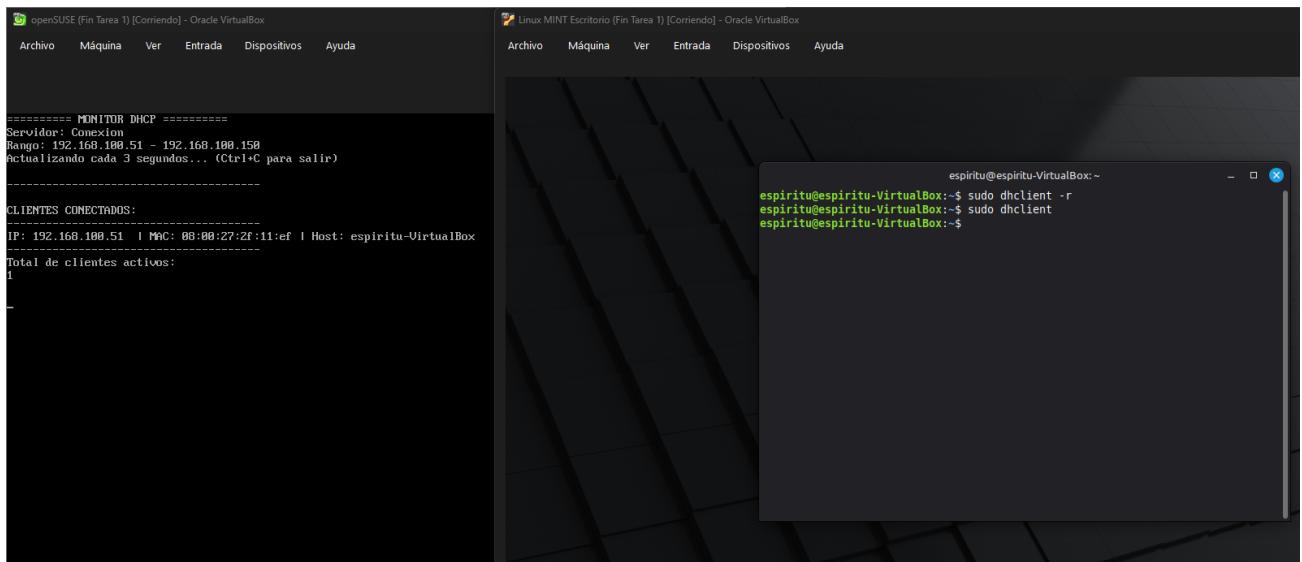
- Evidencias de Configuración:

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)



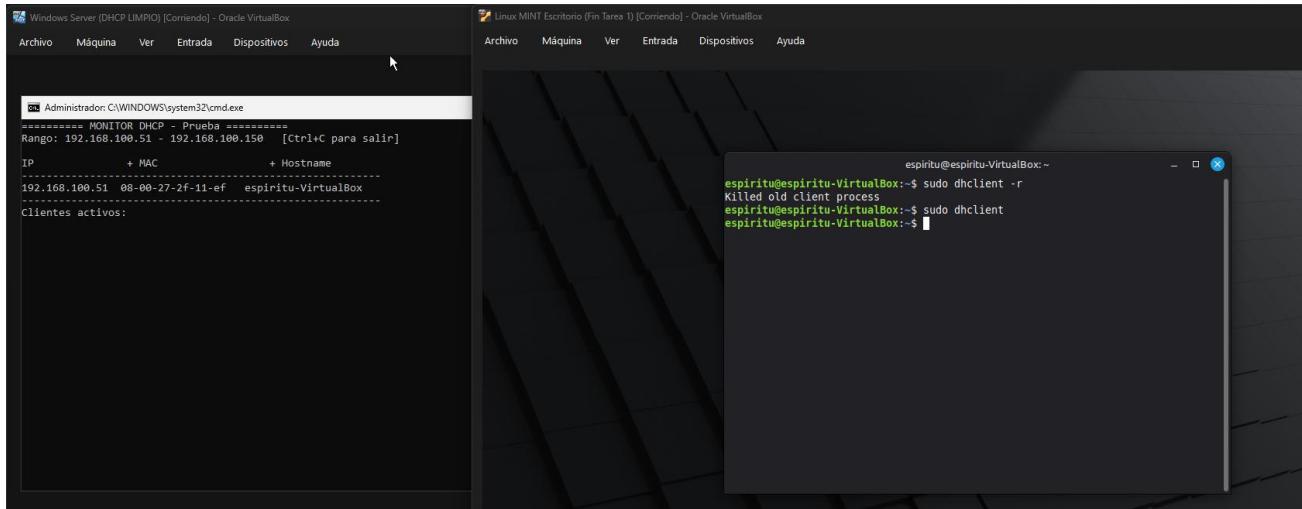
3- Asignación manual de tarjeta de red:

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

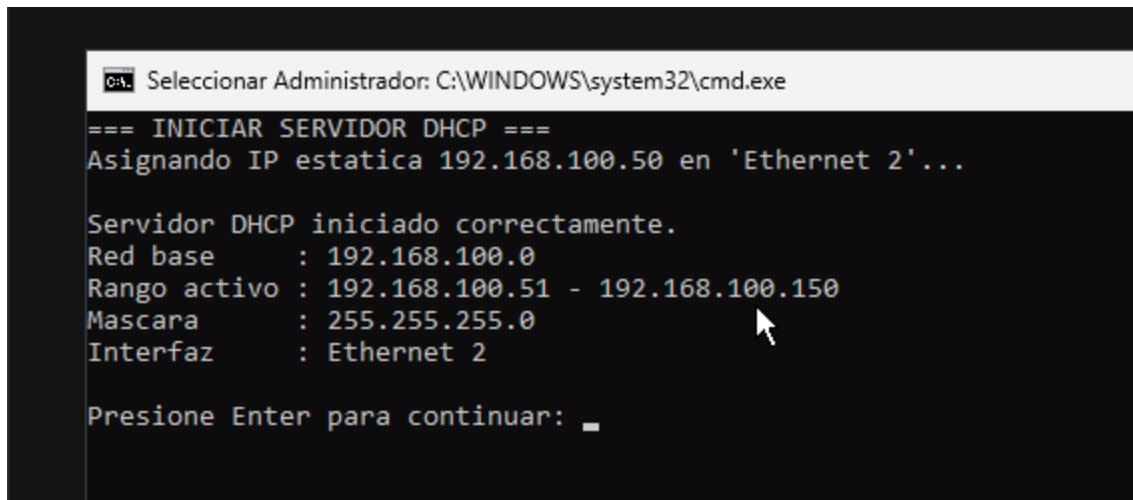
## Path:      Network/DHCP/DHCP server
## Description: DHCPv6 server settings
## Type:      string
## Default:    ""
## ServiceRestart: dhcpcd6
#
# Interface(s) for the DHCPv6 server to listen on.
#
# A special keyword is ANY, it will cause dhcpcd to autodetect available
# interfaces.
#
```

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, demás ahora tengo la idea de como es la estructura de la configuración de un servidor dhcp. Además, no sabia como agregar el dns secundario a la generación del servidor, además se me complicó asignarle una ip estatica a openSUSE.

- 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/>