



Universidad Autónoma de Sinaloa

Licenciatura en Ingeniería de Software



Tarea 1: Entorno de Virtualización e Infraestructura Base

Materia:

Administración de Sistemas

Nombre del Profesor:

Dr. Herman Geovany Ayala Zuñiga

Nombre del alumno:

Espiritu Manzanarez Diego Manuel

Fecha:

02/02/2026

➤ Introducción y Arquitectura o lo que trate cada práctica

- Historial de Cambios:

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

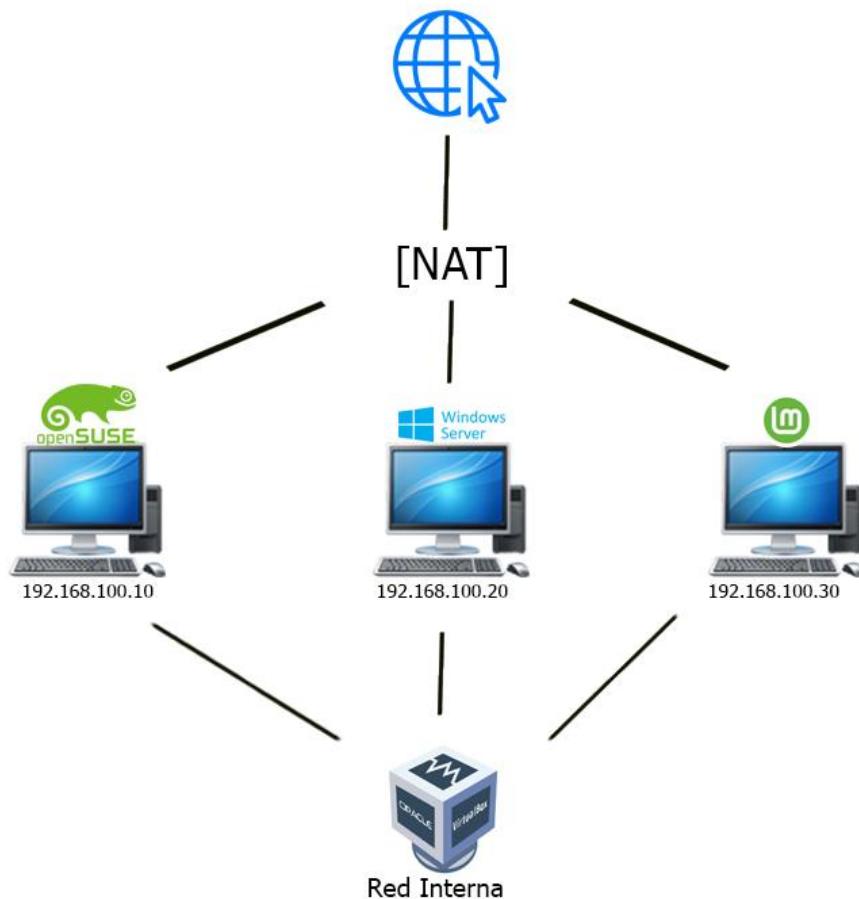
Versión	Fecha	Descripción del Cambio	Relación con Git
1.0	02/02/2026	Se crearon las carpetas donde se guardara el contenido de la tarea.	Tarea 1: Carpetas Creadas
1.1	02/02/2026	Se agrego el archivo README principal.	Tarea 1: Agregar README
1.2	02/02/2026	Se confirmaron los datos de README y las carpetas.	Tarea 1: Confirmar carpetas
1.3	02/02/2026	Se aplicaron IPs Estaticas a las maquinas y se hizo ping entre ellas, se guardaron capturas de evidencia.	Tarea 1: Ping entre maquinas
1.4	02/02/2026	Se tomaron las Snapshots en VirtualBox en el momento de asignar las ips.	Tarea 1: Snapshots tomadas con las ips asignadas
1.5	02/02/2026	Se subieron las capturas donde se crea y ejecutan los scritps (.sh y .ps1).	Tarea 1: Creacion y Ejecucion de Scripts
1.6	02/02/2026	Se subio a github el archivo sh desde la maquina openSUSE	Tarea 1: Guardar Script sh
1.7	02/02/2026	Se subio a github el archivo ps1 desde la maquina Windows Server	Tarea 1: Guardar script ps1
1.8	02/02/2026	Se restructuro la carpeta y se adjunto el documento final.	Tarea 1: Documento subido

- Objetivo:

Montar un entorno de red virtualizado utilizando 3 nodos garantizando la comunicación interna, el aislamiento de red y la preparación de los sistemas operativos para las automatizaciones, los 'SO' utilizados son:

- openSUSE Leap 16.0
- Windows Server 2025 Standard Edition
- Linux Mint

- Diagrama de Topología:



➤ Guía de Uso de los Scripts

- Requisitos Previos:

- Crear las máquinas virtuales
- Tener configurados los adaptadores tanto NAT como de red interna
- Asignar direcciones IP a cada maquina
- Tener git descargado y relacionado con el repositorio de github

- Instrucciones de Ejecución:

Archivo Sh (openSUSE)

1. Primero se crea el archivo

```
Despiritu@Srv-Linux-Sistemas:~> touch Check_Status.sh
Despiritu@Srv-Linux-Sistemas:~> ls
Check_Status.sh
Despiritu@Srv-Linux-Sistemas:~> sudo nano Check_Status.sh
[sudo] contraseña para Despiritu:
```

2. Con el sudo nano agregamos los comandos necesarios

```
echo -e "\nNombre del equipo: $(hostname)\n"
```

```
echo "Ip:"
```

```
hostname -I
```

```
echo -e "\nEspacio en Disco:"
```

```
df -h
```

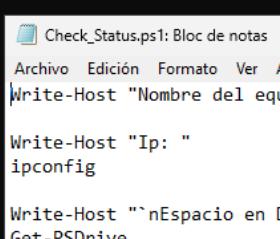
3. Ejecutamos el archivo con el comando “sh”

```
Despiritu@Srv-Linux-Sistemas:~> sh Check_Status.sh
Nombre del equipo: Srv-Linux-Sistemas
Ip:
10.0.2.15 192.168.100.10 fd17:625c:f037:2:128e:be38:568c:39b9 fd17:625c:f037:2:470e:521d:a848:4262
Espacio en Disco:
S.ficheros    Tamaño Usados  Disp Uso% Montado en
/dev/sda2      38G   3.8G  34G  10% /
devtmpfs       4.0M     0  4.0M  0% /dev
tmpfs          2.0G     0  2.0G  0% /dev/shm
tmpfs          784M  728K  783M  1% /run
tmpfs          1.0M     0  1.0M  0% /run/credentials/systemd-journald.service
/dev/sda2      38G   3.8G  34G  10% /.snapshots
/dev/sda2      38G   3.8G  34G  10% /boot/grub2/x86_64-efi
/dev/sda2      38G   3.8G  34G  10% /boot/grub2/i386-pc
/dev/sda2      38G   3.8G  34G  10% /home
/dev/sda2      38G   3.8G  34G  10% /root
/dev/sda2      38G   3.8G  34G  10% /opt
/dev/sda2      38G   3.8G  34G  10% /srv
/dev/sda2      38G   3.8G  34G  10% /usr/local
tmpfs          2.0G     0  2.0G  0% /tmp
/dev/sda2      38G   3.8G  34G  10% /var
tmpfs          1.0M     0  1.0M  0% /run/credentials/getty@tty1.service
tmpfs          392M  4.0K  392M  1% /run/user/1000
Despiritu@Srv-Linux-Sistemas:~>
```

Archivo ps1 (Windows Server)

1. Se crea el archivo ps1 y se escribe dentro el código a ejecutar

```
C:\> Administrador: C:\WINDOWS\system32\cmd.exe
PS C:\Users\Administrador> notepad .\Check_Status.ps1
PS C:\Users\Administrador>



```
Check_Status.ps1: Bloc de notas

Archivo Edición Formato Ver Ayuda
Write-Host "Nombre del equipo: $env:COMPUTERNAME `n"
Write-Host "Ip: "
ipconfig

Write-Host "`nEspacio en Disco: "
Get-PSDrive
```


```

2. Se ejecuta el archivo ps1

➤ Bitácora de Desarrollo y Configuración

- Explicación del Script:

Archivo sh (openSUSE)

```
echo -e "\nNombre del equipo: $(hostname)\n"
```

```
echo "Ip:"
```

```
hostname -I
```

```
echo -e "\nEspacio en Disco:"
```

```
df -h
```

Con el comando “echo” mostramos en pantalla “Nombre del equipo”, de adjuntamos el parámetro “-e”, este nos permite usar el salto de línea “\n”, ademas se le agrego “\$(hostname)” esto mostrara el nombre del equipo dentro del texto mostrado en pantalla.

Hostname -I nos muestra la IP existente en el equipo.

El comando “df” nos muestra el espacio del disco, se le agrega el parámetro “-h” el cual muestra de forma legible la capacidad del disco mostrándola en mg, gb, etc.

Archivo ps1 (Windows Server)

```
Write-Host "Nombre del equipo: $env:COMPUTERNAME `n"
```

```
Write-Host "Ip: "
```

```
ipconfig
```

```
Write-Host "`nEspacio en Disco: "
```

```
Get-PSDrive
```

Con el comando “Write-Host” se muestra la información directamente en pantalla. En la primera línea se imprime el texto “Nombre del equipo” y se concatena la variable de entorno “\$env:COMPUTERNAME”, la cual contiene el nombre del equipo en Windows.

Se agrega `n para realizar un salto de línea.

Posteriormente, se utiliza “Write-Host” para mostrar el texto “Ip:” y a continuación se ejecuta el comando “ipconfig”, el cual muestra la configuración de red del equipo, incluyendo las direcciones IP disponibles.

Finalmente, se imprime el texto “Espacio en Disco” usando “Write-Host” junto con un salto de línea, y se ejecuta el comando “Get-PSDrive”, el cual muestra las unidades del sistema y su espacio disponible, permitiendo visualizar el uso del disco en el equipo.

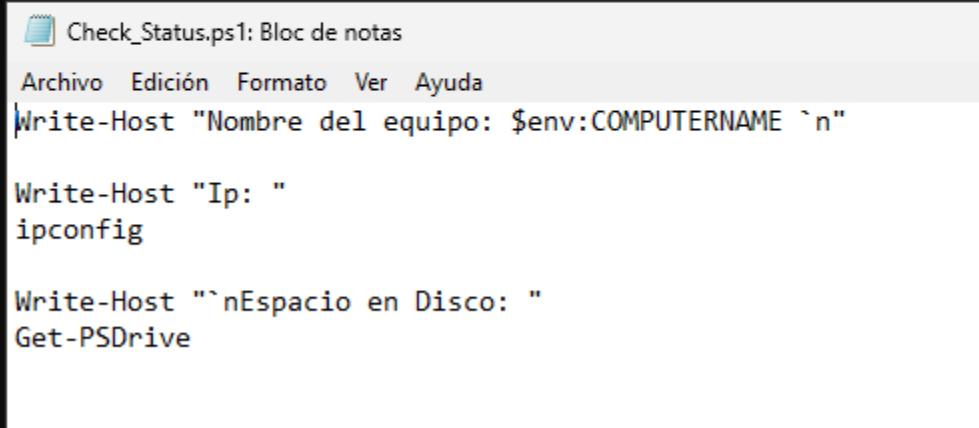
- Evidencias de Configuración:

Scripts:

```
GNU nano 8.6
echo -e "\nNombre del equipo: $(hostname)\n"
echo "Ip:"
hostname -I

echo -e "\nEspacio en Disco:"
df -h
```

```
ca Administrador: C:\WINDOWS\system32\cmd.exe
PS C:\Users\Administrador> notepad .\Check_Status.ps1
PS C:\Users\Administrador>



```
Check_Status.ps1: Bloc de notas
Archivo Edición Formato Ver Ayuda
Write-Host "Nombre del equipo: $env:COMPUTERNAME `n"

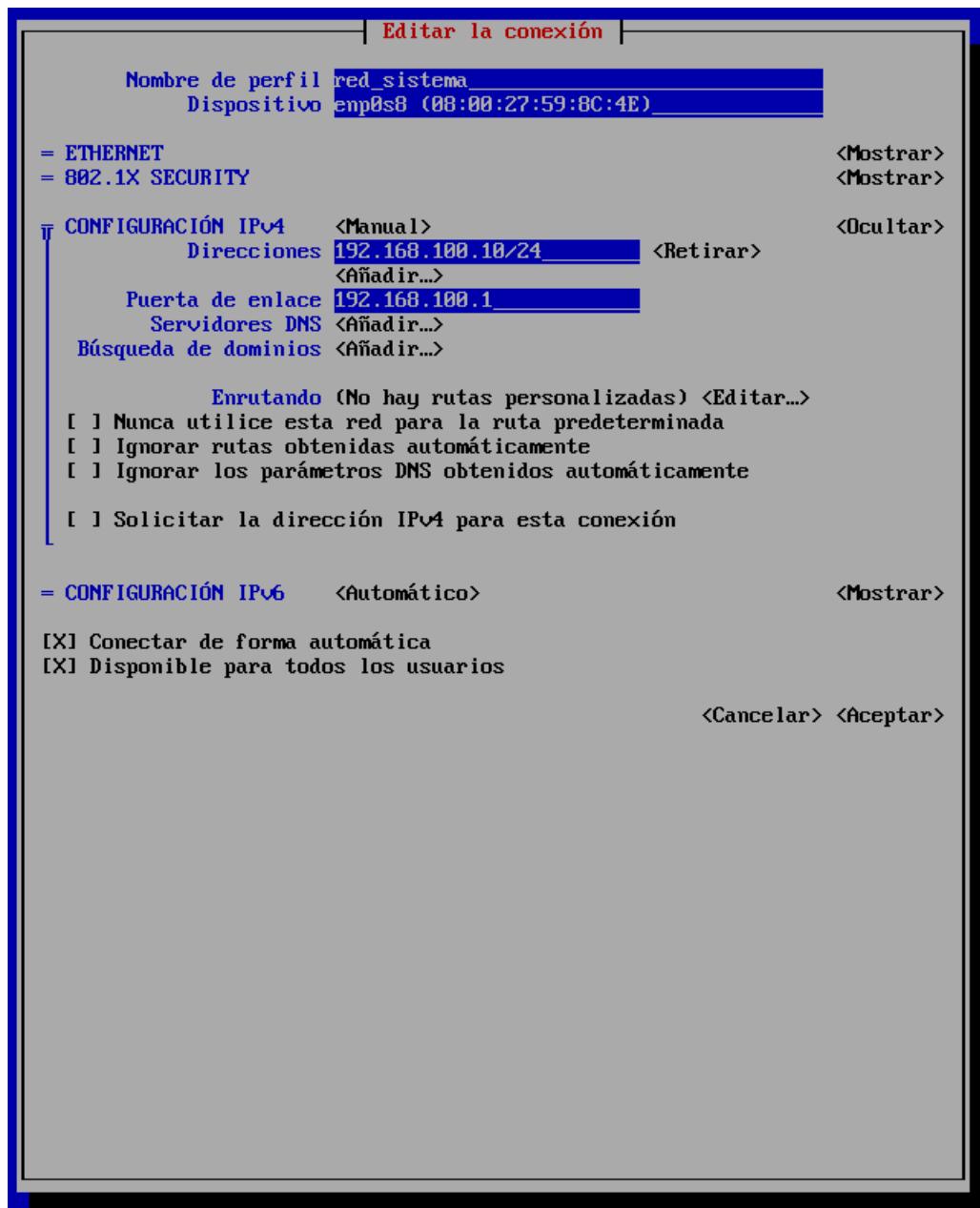
Write-Host "Ip: "
ipconfig

Write-Host "`nEspacio en Disco: "
Get-PSDrive
```


```

IPs Estáticas:

openSUSE



Windows Server

```
sc SConfig: Windows Server 2025 Standard Evaluation, WIN-N0A2O6552VO
=====
Configuración de adaptador de red
=====

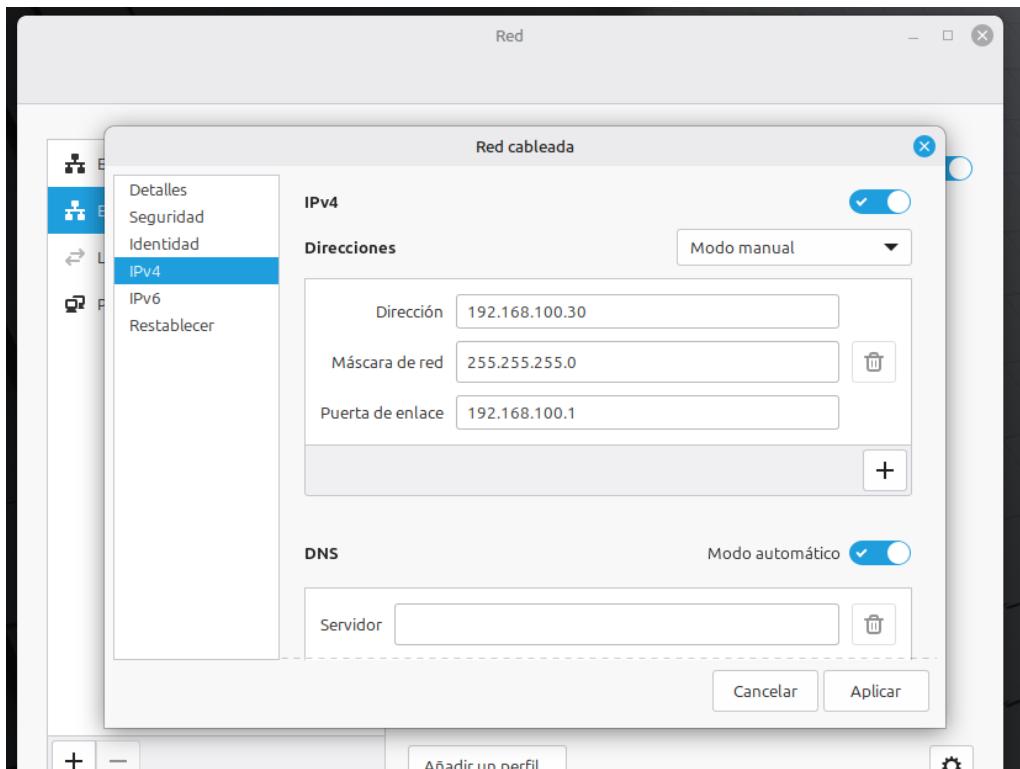
Índice NIC:      7
Nombre:          Ethernet 2
Descripción:    Intel(R) PRO/1000 MT Desktop Adapter #2
Dirección IP:   192.168.100.20,
                fe80::d774:bef5:ca98:2d3b
Máscara de subred: 255.255.255.0
DHCP habilitado: False

Puerta de enlace predeterminada: 192.168.100.1
Primer servidor DNS:
Segundo servidor DNS:
Tercer servidor DNS:

1) Establecer dirección del adaptador de red
2) Establecer servidores DNS
3) Borrar configuración de servidores DNS
4) Cambiar el nombre del adaptador de red

Escribir selección (En blanco=Cancelar):
```

Linux Mint



➤ Pruebas de Funcionamiento (Validación):

- Protocolo de Pruebas:

Archivo sh (openSUSE)

Entrada	Salida Esperada	Salida Obtenida
---	Nombre del equipo Ip Espacio en disco	Srv-Linux-Sistemas 192.168.100.10 [Espacio en disco]

Archivo ps1 (Windows Server)

Entrada	Salida Esperada	Salida Obtenida
---	Nombre del equipo Ip Espacio en disco	Srv-Win-Sistema 192.168.100.20 [Espacio en disco]

- Capturas de Validación:

Capturas de Scripts:

```
Despiritu@Srv-Linux-Sistemas:~> sh Check_Status.sh
Nombre del equipo: Srv-Linux-Sistemas
Ip:
10.0.2.15 192.168.100.10 fd17:625c:f037:2:128e:be38:568c:39b9 fd17:625c:f037:2:470e:521d:a848:4262

Espacio en Disco:
S.ficheros Tamaño Usados Disp Uso% Montado en
/dev/sda2 38G 3.8G 34G 10% /
devtmpfs 4.0M 0 4.0M 0% /dev
tmpfs 2.0G 0 2.0G 0% /dev/shm
tmpfs 784M 728K 783M 1% /run
tmpfs 1.0M 0 1.0M 0% /run/credentials/systemd-journald.service
/dev/sda2 38G 3.8G 34G 10% /.snapshots
/dev/sda2 38G 3.8G 34G 10% /boot/grub2/x86_64-efi
/dev/sda2 38G 3.8G 34G 10% /boot/grub2/i386-pc
/dev/sda2 38G 3.8G 34G 10% /home
/dev/sda2 38G 3.8G 34G 10% /root
/dev/sda2 38G 3.8G 34G 10% /opt
/dev/sda2 38G 3.8G 34G 10% /srv
/dev/sda2 38G 3.8G 34G 10% /usr/local
tmpfs 2.0G 0 2.0G 0% /tmp
/dev/sda2 38G 3.8G 34G 10% /var
tmpfs 1.0M 0 1.0M 0% /run/credentials/getty@tty1.service
tmpfs 392M 4.0K 392M 1% /run/user/1000
Despiritu@Srv-Linux-Sistemas:>
```

```
Administrator: C:\WINDOWS\system32\cmd.exe
PS C:\Users\Administrador> notepad .\Check_Status.ps1
PS C:\Users\Administrador> .\Check_Status.ps1
Nombre del equipo: SRV-WIN-SISTEMA

Ip:

Configuración IP de Windows

Adaptador de Ethernet Ethernet:
Sufijo DNS específico para la conexión. . :
Dirección IPv6 . . . . . : fd17:625c:f037:2:eb9b:2ffa:86d7:a2b6
Vínculo: dirección IPv6 local. . . : fe80::a4f3:fb7e:143:f681%
Dirección IPv4. . . . . : 10.0.2.15
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . : fe80::2%4
10.0.2.2

Adaptador de Ethernet Ethernet 2:
Sufijo DNS específico para la conexión. . :
Vínculo: dirección IPv6 local. . . : fe80::d774:bef5:ca98:2d3b%7
Dirección IPv4. . . . . : 192.168.100.20
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . : 192.168.100.1

Espacio en Disco:
Name Used (GB) Free (GB) Provider Root
---- ----- ----- -----
Alias Alias
C 9,66 39,46 FileSystem C:\
Cert Certificate \
Env Environment
Function Function
HKCU Registry HKEY_CURRENT_USER
HKLM Registry HKEY_LOCAL_MACHINE
Variable Variable
WSMan WSMAN

PS C:\Users\Administrador> -
```

Capturas de Ping entre maquinas:

openSUSE

```
Despiritu@Srv-Linux-Sistemas:~> ping -c 4 192.168.100.20
PING 192.168.100.20 (192.168.100.20) 56(84) bytes of data.
64 bytes from 192.168.100.20: icmp_seq=1 ttl=128 time=1.45 ms
64 bytes from 192.168.100.20: icmp_seq=2 ttl=128 time=0.781 ms
64 bytes from 192.168.100.20: icmp_seq=3 ttl=128 time=0.518 ms
64 bytes from 192.168.100.20: icmp_seq=4 ttl=128 time=1.03 ms

--- 192.168.100.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.518/0.946/1.452/0.344 ms
Despiritu@Srv-Linux-Sistemas:~>
Despiritu@Srv-Linux-Sistemas:~>
Despiritu@Srv-Linux-Sistemas:~>
Despiritu@Srv-Linux-Sistemas:~> ping -c 4 192.168.100.30
PING 192.168.100.30 (192.168.100.30) 56(84) bytes of data.
64 bytes from 192.168.100.30: icmp_seq=1 ttl=64 time=1.58 ms
64 bytes from 192.168.100.30: icmp_seq=2 ttl=64 time=0.709 ms
64 bytes from 192.168.100.30: icmp_seq=3 ttl=64 time=0.505 ms
64 bytes from 192.168.100.30: icmp_seq=4 ttl=64 time=0.718 ms

--- 192.168.100.30 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 0.505/0.877/1.578/0.413 ms
Despiritu@Srv-Linux-Sistemas:~> _
```

Windows Server

```
Administrator: C:\WINDOWS\system32\cmd.exe
PS C:\Users\Administrador> ping 192.168.100.10

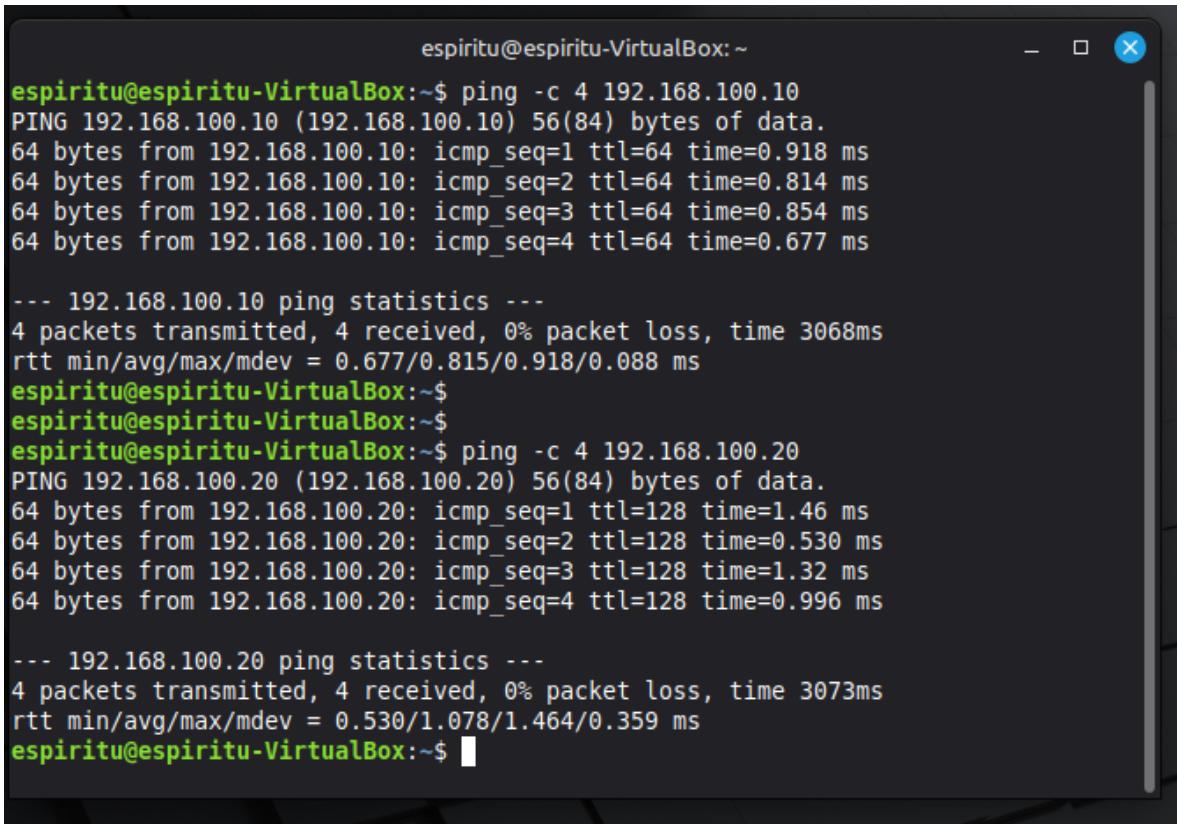
Haciendo ping a 192.168.100.10 con 32 bytes de datos:
Respuesta desde 192.168.100.10: bytes=32 tiempo<1ms TTL=64
Respuesta desde 192.168.100.10: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.100.10: bytes=32 tiempo<1ms TTL=64
Respuesta desde 192.168.100.10: bytes=32 tiempo<1ms TTL=64

Estadísticas de ping para 192.168.100.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms
PS C:\Users\Administrador>
PS C:\Users\Administrador>
PS C:\Users\Administrador> ping 192.168.100.30

Haciendo ping a 192.168.100.30 con 32 bytes de datos:
Respuesta desde 192.168.100.30: bytes=32 tiempo<1ms TTL=64

Estadísticas de ping para 192.168.100.30:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms
PS C:\Users\Administrador>
```

Linux Mint



```
espiritu@espiritu-VirtualBox:~$ ping -c 4 192.168.100.10
PING 192.168.100.10 (192.168.100.10) 56(84) bytes of data.
64 bytes from 192.168.100.10: icmp_seq=1 ttl=64 time=0.918 ms
64 bytes from 192.168.100.10: icmp_seq=2 ttl=64 time=0.814 ms
64 bytes from 192.168.100.10: icmp_seq=3 ttl=64 time=0.854 ms
64 bytes from 192.168.100.10: icmp_seq=4 ttl=64 time=0.677 ms

--- 192.168.100.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3068ms
rtt min/avg/max/mdev = 0.677/0.815/0.918/0.088 ms
espiritu@espiritu-VirtualBox:~$
espiritu@espiritu-VirtualBox:~$ ping -c 4 192.168.100.20
PING 192.168.100.20 (192.168.100.20) 56(84) bytes of data.
64 bytes from 192.168.100.20: icmp_seq=1 ttl=128 time=1.46 ms
64 bytes from 192.168.100.20: icmp_seq=2 ttl=128 time=0.530 ms
64 bytes from 192.168.100.20: icmp_seq=3 ttl=128 time=1.32 ms
64 bytes from 192.168.100.20: icmp_seq=4 ttl=128 time=0.996 ms

--- 192.168.100.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3073ms
rtt min/avg/max/mdev = 0.530/1.078/1.464/0.359 ms
espiritu@espiritu-VirtualBox:~$
```

➤ Conclusiones y Referencias:

- Lecciones Aprendidas:

1. Creación de Tokens en GitHub:

Al momento de querer hacer un “push” desde las maquinas virtuales me solicitaba un usuario y contraseña a pesar de que era un repositorio publico, por lo que me vi forzado a generar un token que me diera acceso a mi repositorio para hacer “push”.

2. Instalacion de Git en Windows Server

Cuando intente descargar git en Windows server desde la consola no me respondia como lo esperaba por lo que cargu el ejecutable de git en una usb y la monte en la maquina virtual para ejecutar el ejecutable y poder descargar el programa.

- Bibliografia:

<https://mundobytes.com/como-asignar-una-ip-estatica-en-windows-desde-cmd-o-powershell/>

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

<https://www.tutorialspoint.com/powershell/index.htm>

<https://www.geeksforgeeks.org/linux-unix/configuring-ip-networking-with-nmtui-in-linux/>