



Universidad Autónoma de Sinaloa

Licenciatura en Ingeniería de Software



Práctica 3: Automatización del Servidor DNS (reprobados.com)

Materia:

Administración de Sistemas

Nombre del Profesor:

Dr. Herman Geovany Ayala Zúñiga

Nombre del alumno:

Espiritu Manzanarez Diego Manuel

Fecha:

25/02/2026

➤ Introducción y Arquitectura o lo que trate cada práctica

- Historial de Cambios:

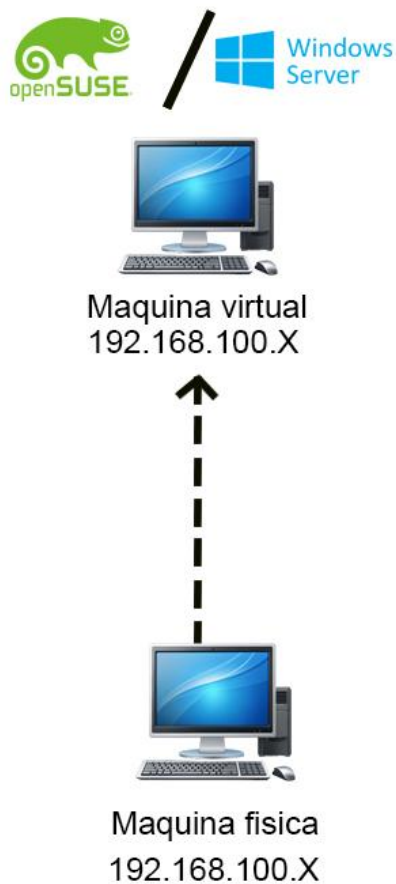
Repositorio: <https://github.com/Diegoesp-0/Administracion-de-Sistemas>

Versión	Fecha	Descripción del Cambio	Relación con Git
4.0	22/02/2026	Creación de carpetas y scripts modulados para probar.	Tarea 4: Carpetas creadas
4.1	22/02/2026	Se corrigen errores en la modularidad de los archivos	Tarea 4: Corrección en los archivos de openSUSE
4.2	22/02/2026	Se corrigieron los scrips de openSUSE para que sean 100% funcionales.	Tarea 4: Scripts openSUSE arreglados
4.3	24/02/2026	Se crean scripts modulados para Windows server,	Tarea 4: Scripts para Windows server
4.4	25/02/2026	Se guardo en archivo ssh.sh de openSUE.	Tarea 4: SSH openSUSE
4.5	25/02/2026	Se guardo el archivo ssh-windows.ps1 de Windows server.	Tarea 4: SSH windows
4.6	25/02/2026	Se guardo el archivo y se guardaron las capturas de evidencia.	Tarea 4: Subida de documentacion

- **Objetivo:**

Acceder a la vm desde la maquina virtual usando el servicio de ssh y poder ejecutar acciones dentro de la vm, además de poder mandar archivos directamente sin tener que utilizar otros métodos como carpetas compartidas, GitHub o cualquier otro método menos eficiente.

- **Diagrama de Topología:**



➤ Guía de Uso de los Scripts

- Requisitos previos:

- Tener una tercera tarjeta de red en puente en la vm.
- Asegurarse de que las maquinas tanto virtual como física hagan ping.
- Tener usuarios preparados en caso de no tenerlos.
- En Windows server crear una regla en el puerto 22 o el que se usara.
- Tener el promisc mode activado en el puente.

➤ Instrucciones de Ejecución:

OpenSUSE

- 1- Ejecutar el comando “sudo bash ssh.sh verificar” esto verificara si el paquete openssh-server esta instalado, en caso de no estar instalado preguntara si desea instalar.

```
OpenSSH-Server esta instalado :D
Despiritu@Srv-Linux-Sistemas:~/Administracion-de-Sistemas/Tarea 4 - SSH/Scripts> _
```

- 2- Ejecutar el comando “sudo bash ssh.sh iniciar”, se iniciara el servicio ssh y en pantalla aparecerá la ip y el puerto que se deben ingresar en PuTTY.

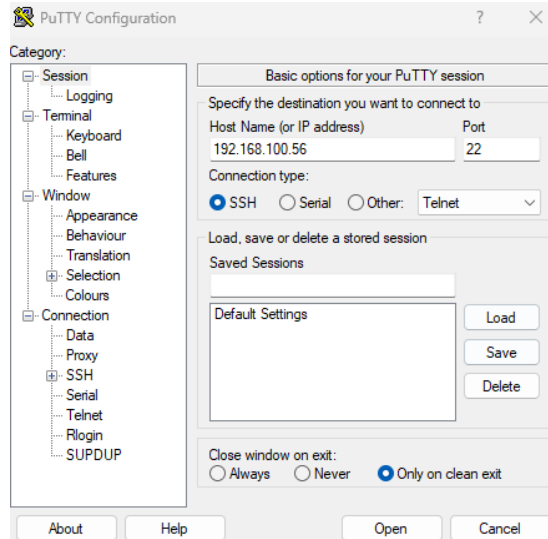
```
success
success

Servidor SSH iniciado correctamente
Puerto 22 abierto en el firewall

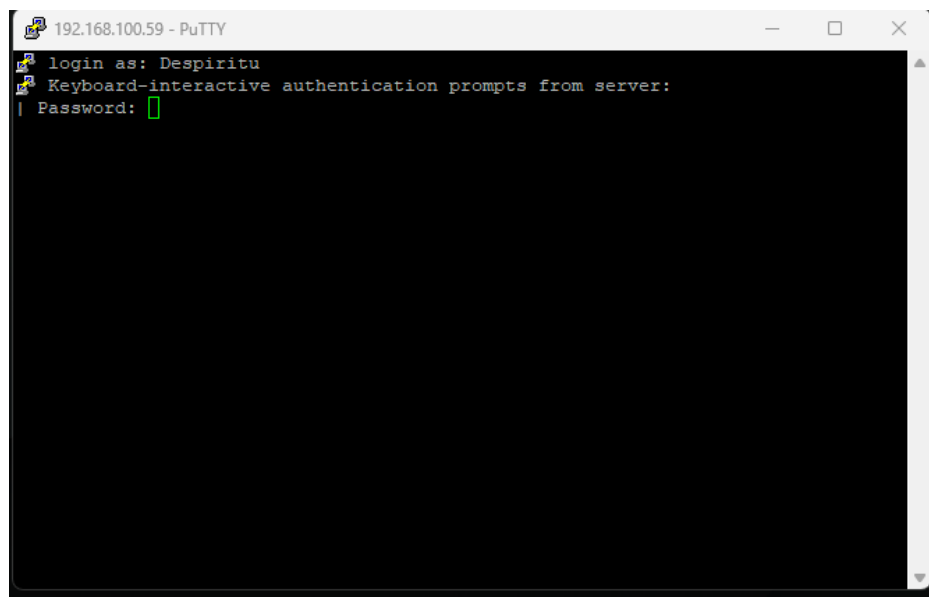
Conectate desde PuTTY:
Host: 192.168.100.56
Port: 22

Despiritu@Srv-Linux-Sistemas:~/Administracion-de-Sistemas/Tarea 4 - SSH/Scripts> _
```

3- Poner los datos que aparecen en consola directamente en el puTTY para conectarse.



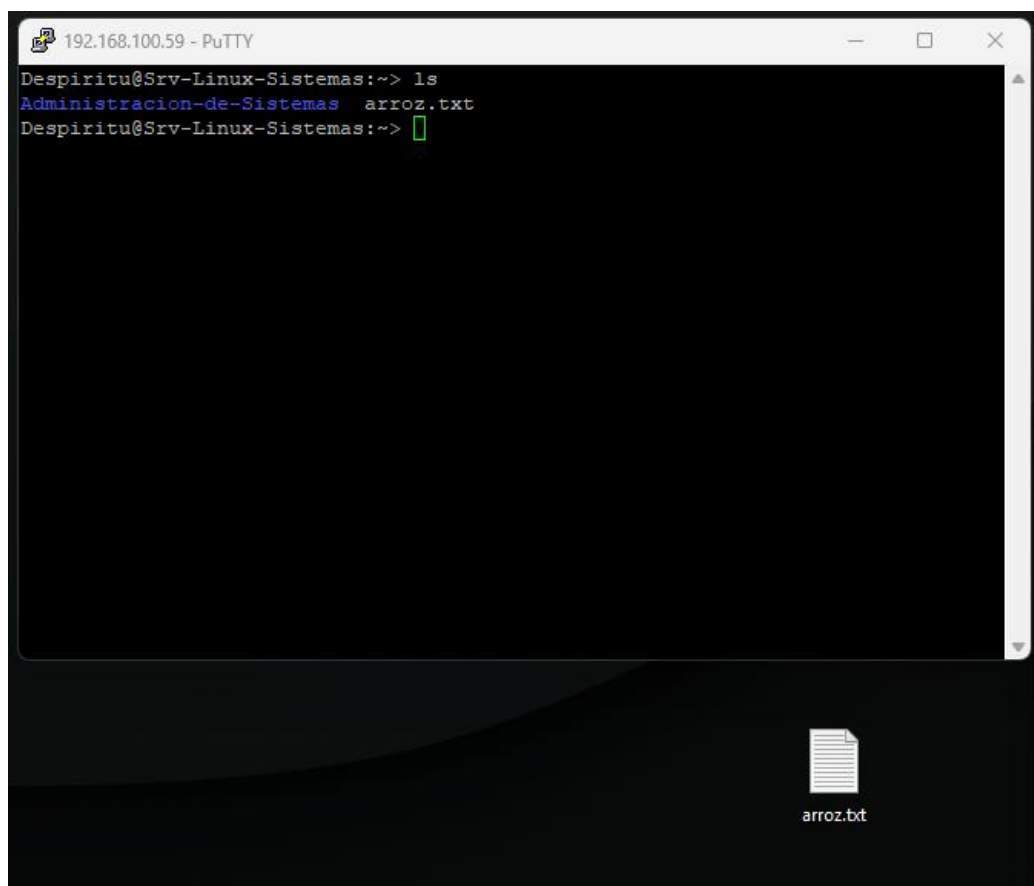
4- En ventana tendrás que iniciar sesión con los datos de usuario de la vm y ya podras usar la vm desde la maquina física.



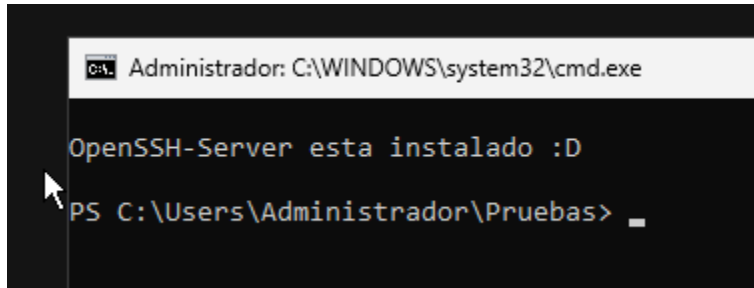
- 5- En una ventana de powershell de la maquina física tenemos que poder el siguiente comando:
scp [ruta del archivo a copiar] [Nombre del usuario de la vm]@[Ip de la vm]:[Ruta donde se pegara], ponemos "yes" y después la contraseña del usuario, y como se ve en la imagen se envió nuestro archivo de prueba "arroz.txt".

```
Windows PowerShell
PS C:\Users\diego\OneDrive\Escritorio\Administracion-de-Sistemas> scp "C:\Users\diego\OneDrive\Escritorio\arroz.txt" Despiritu@192.168.100.59:/home/Despiritu
The authenticity of host '192.168.100.59 (192.168.100.59)' can't be established.
ED25519 key fingerprint is SHA256:OUNgAz8179eW6Vd6rC+AZWCaImHsTRfUMt6l61nc05k.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '192.168.100.59' (ED25519) to the list of known hosts.
(Despiritu@192.168.100.59) Password:
arroz.txt                                100%    0    0.0KB/s    00:00
PS C:\Users\diego\OneDrive\Escritorio\Administracion-de-Sistemas>
```

Windows Server



- 1- Ejecutar el comando `ssh_windows.ps1` verificar, esto comprobara si esta descargada la paquetería para ssh y en caso de que no, se podrá instalar.

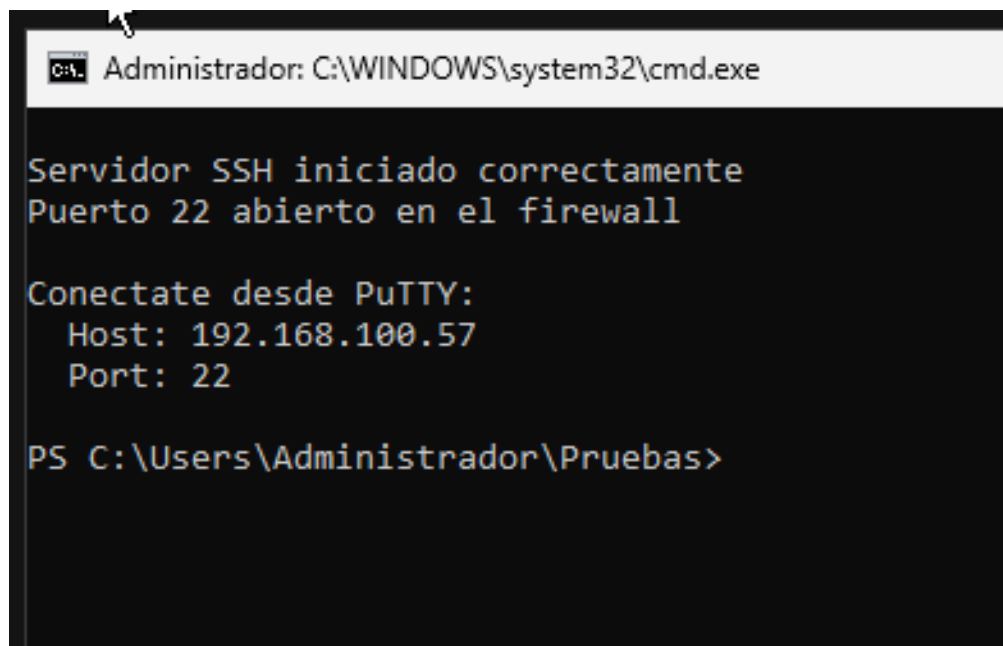


```
Administrador: C:\WINDOWS\system32\cmd.exe

OpenSSH-Server esta instalado :D

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

- 2- Ejecutar el comando `ssh_windows.ps1` iniciar, esto iniciara el servicio y mostrara los datos a ingresar en PuTTY.



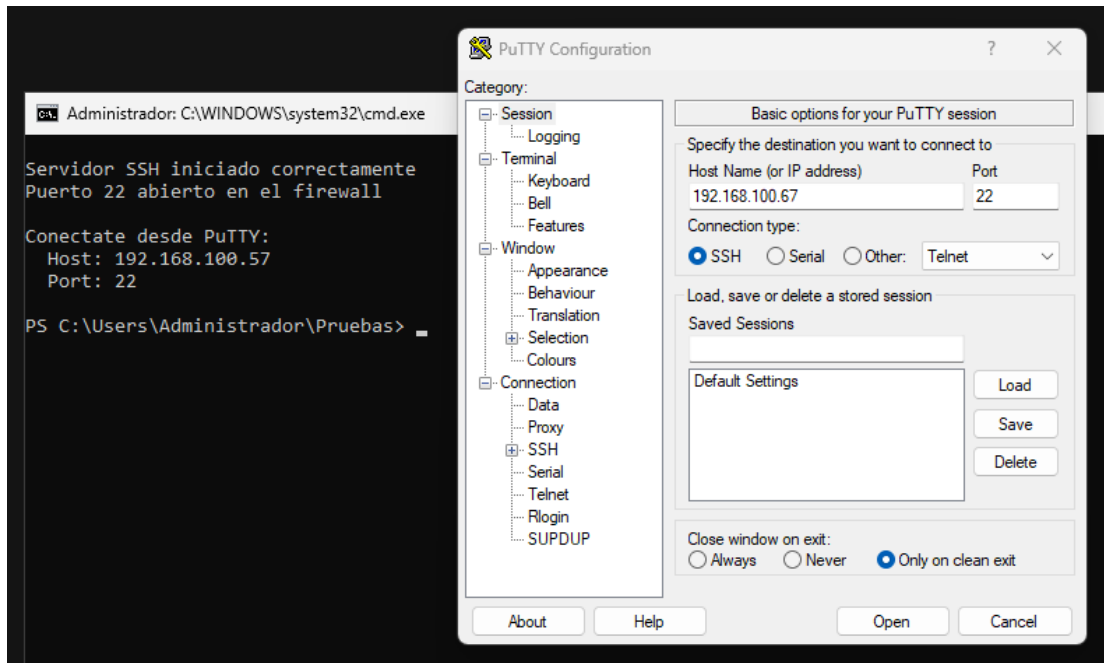
```
Administrador: C:\WINDOWS\system32\cmd.exe

Servidor SSH iniciado correctamente
Puerto 22 abierto en el firewall

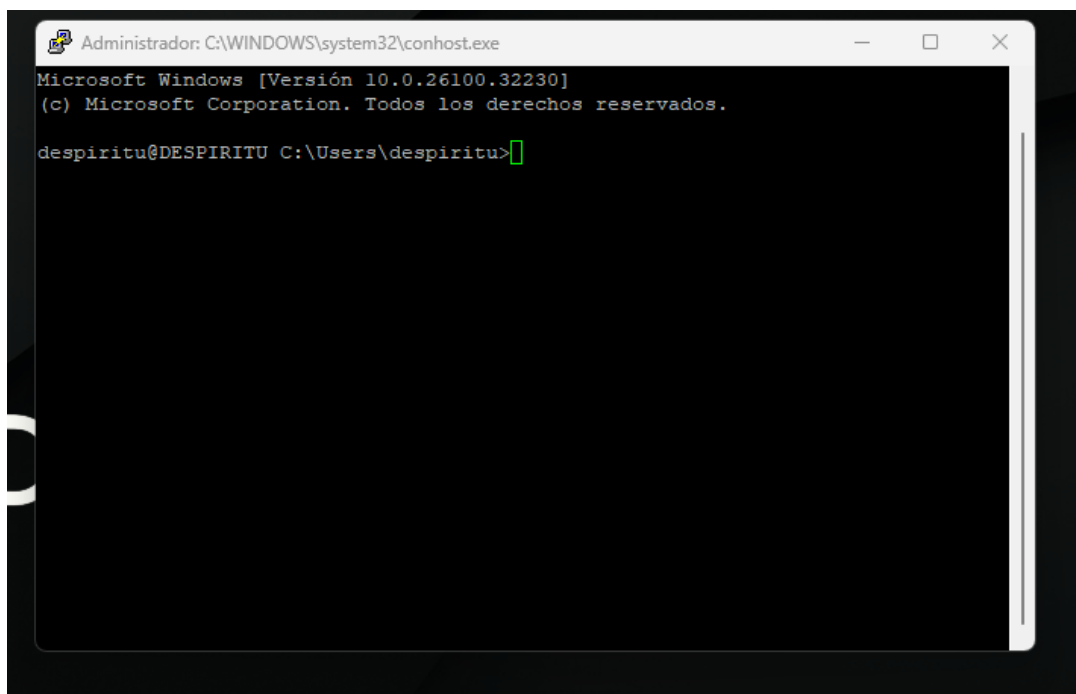
Conectate desde PuTTY:
  Host: 192.168.100.57
  Port: 22

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

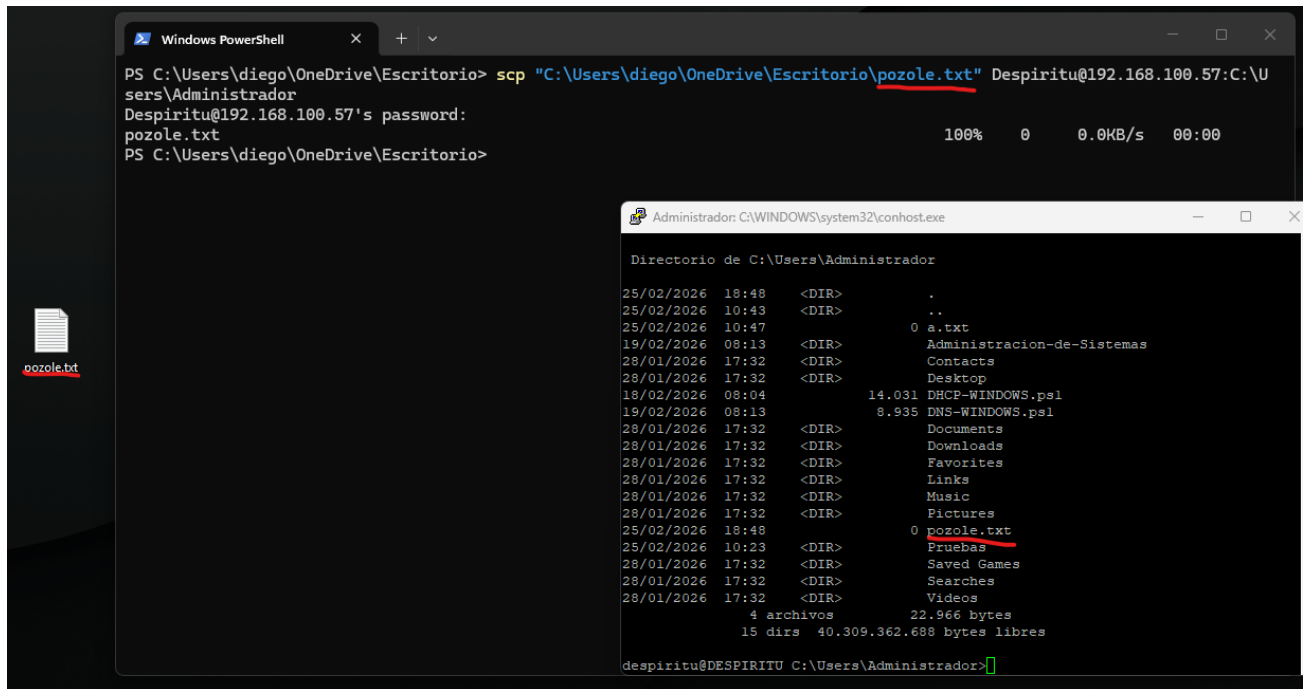
3- Colocar los datos que aparecen en la consola en puTTY.



4- Escribir el usuario y contraseña del usuario de Windows para acceder.



5- En una ventana de powershell de la maquina física tenemos que poder el siguiente comando:
scp [ruta del archivo a copiar] [Nombre del usuario de la vm]@[ip de la vm]:[Ruta donde se pegara], ponemos "yes" y después la contraseña del usuario, y como se ve en la imagen se envió nuestro archivo de prueba "pozle.com".



```
Windows PowerShell
PS C:\Users\diego\OneDrive\Escritorio> scp "C:\Users\diego\OneDrive\Escritorio\pozole.txt" Despiritu@192.168.100.57:C:\Users\Administrador
Despiritu@192.168.100.57's password:
pozole.txt
PS C:\Users\diego\OneDrive\Escritorio>
```

100% 0 0.0KB/s 00:00

Administrador: C:\WINDOWS\system32\conhost.exe

Directorio de C:\Users\Administrador

Fecha y hora	Permisos	Nombre
25/02/2026 18:48	<DIR>	.
25/02/2026 10:43	<DIR>	..
25/02/2026 10:47		0 a.txt
19/02/2026 08:13	<DIR>	Administracion-de-Sistemas
28/01/2026 17:32	<DIR>	Contacts
28/01/2026 17:32	<DIR>	Desktop
18/02/2026 08:04		14.031 DHCP-WINDOWS.ps1
19/02/2026 08:13		8.935 DNS-WINDOWS.ps1
28/01/2026 17:32	<DIR>	Documents
28/01/2026 17:32	<DIR>	Downloads
28/01/2026 17:32	<DIR>	Favorites
28/01/2026 17:32	<DIR>	Links
28/01/2026 17:32	<DIR>	Music
28/01/2026 17:32	<DIR>	Pictures
25/02/2026 18:48		0 pozole.txt
25/02/2026 10:23	<DIR>	Pruebas
28/01/2026 17:32	<DIR>	Saved Games
28/01/2026 17:32	<DIR>	Searches
28/01/2026 17:32	<DIR>	Videos
		4 archivos 22.966 bytes
		15 dirs 40.309.362.688 bytes libres

despiritu@DESPIRITU C:\Users\Administrador>

➤ Bitácora de Desarrollo y Configuración:

- Explicación de Script:
openSUSE

Al ejecutarse, el script activa el modo promiscuo en la interfaz del puente, en este caso la enp0s9.

Función verificar Utiliza rpm -q para consultar la base de datos de paquetes RPM y determinar si openssh-server está instalado. Si el paquete no existe, ofrece instalarlo interactivamente mediante zypper.

Luego verifica si SSH está instalado y, de estarlo, lo habilita con systemctl enable para que arranque automáticamente, lo inicia, y abre el puerto 22 en el firewall con firewall-cmd. Finalmente, extrae la IP de enp0s9 y la muestra como datos de conexión para PuTTY.

Función reiniciar Ejecuta systemctl restart sshd y valida el resultado con systemctl is-active para confirmar que el servicio quedó activo tras el reinicio.

Función detener Detiene el servicio con systemctl stop sshd y evalúa el código de retorno para ver si se detuvo o no.

Windows Server

Función Verificar-SSH Utiliza `Get-WindowsCapability` para consultar el estado de la característica opcional `OpenSSH.Server` directamente desde el sistema operativo Windows. Evalúa la propiedad `State`, si es "Installed" confirma la instalación.

Función Iniciar-SSH Primero verifica que SSH esté instalado con la misma lógica de `Get-WindowsCapability`. Si está disponible, configura el servicio `sshd` como automático usando `Set-Service -StartupType Automatic` para que persista entre reinicios, luego lo inicia con `Start-Service`.

Para el firewall, consulta si ya existe una regla llamada `OpenSSH-Server-In-TCP` con `Get-NetFirewallRule`; solo si no existe, la crea con `New-NetFirewallRule` permitiendo tráfico TCP entrante en el puerto 22.

Función Detener-SSH Llama a `Stop-Service sshd` y de igual forma consulta el estado posterior del servicio.

Función Estado-SSH Obtiene el objeto del servicio con `Get-Service` y lo formatea con `Format-List` mostrando nombre, estado y tipo de inicio.

- Evidencias de Configuración:

OpenSUSE:

```
Despiritu@Srv-Linux-Sistemas:/etc/ssh> ls  
ssh_config.d  sshd_config.d  ssh_host_ecdsa_key  
Despiritu@Srv-Linux-Sistemas:/etc/ssh>
```

Aquí podemos ver el archivo ssh_config.d

```
Despiritu@Srv-Linux-Sistemas:/> systemctl status sshd  
● sshd.service - OpenSSH Daemon  
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: disabled)  
   Active: active (running) since Wed 2026-02-25 11:03:51 MST; 2min 54s ago  
 Invocation: ad6fc667bc6246b8a33267bb83958188  
   Process: 1060 ExecStartPre=/usr/sbin/sshd-gen-keys-start (code=exited, status=0/SUCCESS)  
   Process: 1093 ExecStartPre=/usr/sbin/sshd -t $SSHD_OPTS (code=exited, status=0/SUCCESS)  
  Main PID: 1131 (sshd)  
    Tasks: 1  
     CPU: 224ms  
   CGroup: /system.slice/sshd.service  
           └─1131 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Warning: some journal files were not opened due to insufficient permissions.  
Despiritu@Srv-Linux-Sistemas:/>
```

Aquí se ve el servicio activo

Windows Server:

```
Administrador: C:\WINDOWS\system32\cmd.exe
PS C:\Users\Administrador> Get-WindowsCapability -Online -Name OpenSSH*

Name       : OpenSSH.Client~~~~0.0.1.0
State      : Installed
DisplayName : Cliente OpenSSH
Description : Cliente Secure Shell (SSH) basado en OpenSSH, para la administración segura de claves y el acceso a
              equipos remotos.
DownloadSize : 0
InstallSize  : 0

Name       : OpenSSH.Server~~~~0.0.1.0
State      : Installed
DisplayName : Servidor OpenSSH
Description : Servidor Secure Shell (SSH) basado en OpenSSH, para la administración segura de claves y el acceso
              desde equipos remotos.
DownloadSize : 0
InstallSize  : 0
```

```
Administrador: C:\WINDOWS\system32\cmd.exe
PS C:\Users\Administrador> Get-Service -Name sshd | Format-List Name, Status, StartType

Name       : sshd
Status      : Running
StartType   : Automatic

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

```
Administrador: C:\WINDOWS\system32\cmd.exe
PS C:\ProgramData\ssh> ls

Directorio: C:\ProgramData\ssh

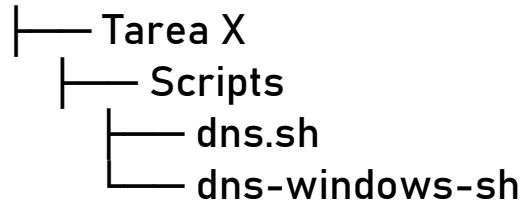
Mode                LastWriteTime         Length Name
----                -
d-----          25/02/2026   10:23             logs
-a----          25/02/2026   11:11              6 sshd.pid
-a----          31/03/2024   18:08          2343 sshd_config
-a----          25/02/2026   10:24          513 ssh_host_ecdsa_key
-a----          25/02/2026   10:24          185 ssh_host_ecdsa_key.pub
-a----          25/02/2026   10:24          419 ssh_host_ed25519_key
-a----          25/02/2026   10:24          105 ssh_host_ed25519_key.pub
-a----          25/02/2026   10:24          2610 ssh_host_rsa_key
-a----          25/02/2026   10:24          577 ssh_host_rsa_key.pub

PS C:\ProgramData\ssh>
```

➤ Arquitectura de Software:

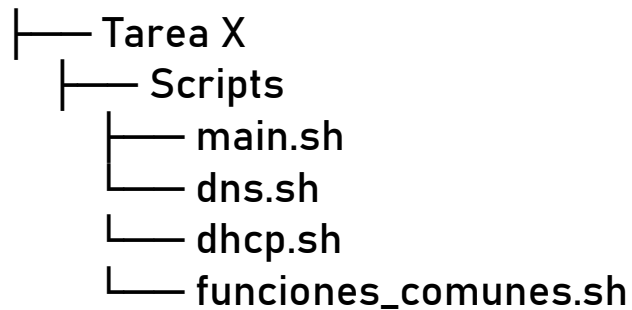
Antes la estructura era algo parecido a esto:

Administracion-de-Sistemas



Ahora la estructura es así:

Administracion-de-Sistemas



Ahora están moduladas en 4 archivos, en el main se abre un menu donde seleccionaras que configurar, si dns o dhcp, dentro de cada uno podrás hacer sus configuraciones, y el archivo funciones_comunes contiene funciones que usan los 2 archivos como el de validar ip y comprobar si esta descargado el paquete que se le envíe a evaluar.

➤ Conclusiones y Referencias

- Lecciones Aprendidas:

Aprendí que era el promisc mode, además de como cambiar el orden de las tarjetas para poder recibir ping, ya que si no están ordenadas el ping nunca llega por el puente lo que significa que la conexión del ssh tampoco podrá funcionar, además a saber que hace cada tarjeta de red y saber usarlas de una mejor manera, también aprendí como enviar archivos a las vm sin tener que usar GitHub u otro medio para poder hacerlo.

- Bibliografía:

<https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units-es>

<https://computernewage.com/2019/09/20/linux-terminal-servicios-gestion/>