



**UNIVERSIDAD AUTÓNOMA DE SINALOA: FACULTAD DE INGENIERÍAS
LOS MOCHIS**

MATERIA: ADMINISTRACIÓN DE SISTEMAS

**PRÁCTICA 1: ENTORNO DE VIRTUALIZACIÓN E INFRAESTRUCTURA
BASE**

FECHA: 09/02/2026

ESTUDIANTE: JOSÉ ALDAIR GARCÍA VALDEZ

REPOSITORIO GITHUB: [ADMINISTRACION-DE-SISTEMAS.GIT](#)

HISTORIAL DE CAMBIOS:

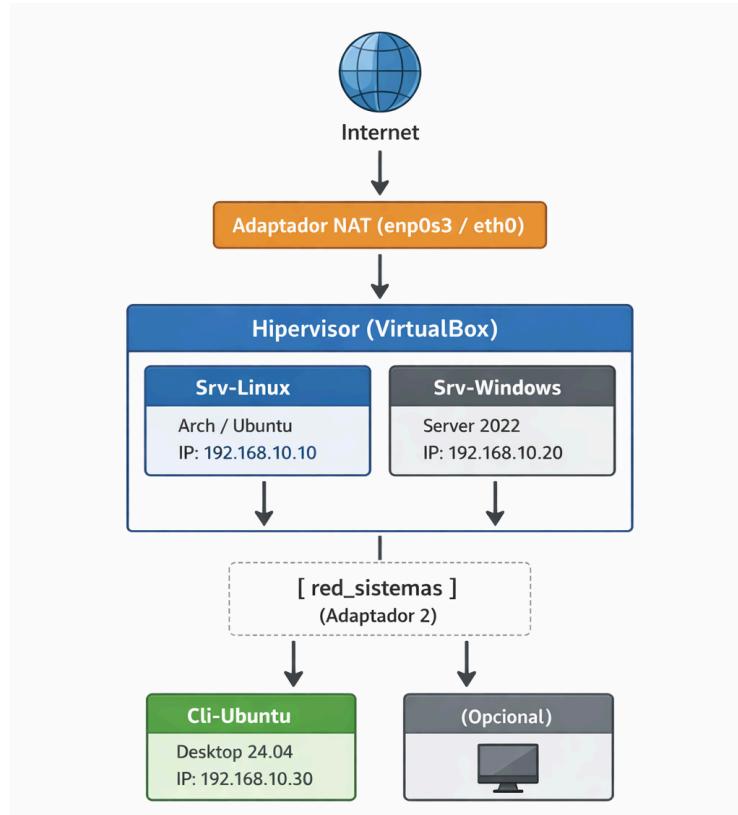
| Versión | Fecha | Cambios Realizados | Commit Hash (GitHub) |
|---------|------------|--|----------------------|
| 1.0 | 09/02/2026 | Creación del entorno virtual e infraestructura base. | ac5faf1 |
| 1.1 | 09/02/2026 | Reorganice la estructura de archivos. | 3c26ec6 |
| 1.2 | 09/02/2026 | Subida de archivo .gitignore. | 7777689 |

Introducción y Arquitectura

Objetivo

Configurar un entorno virtualizado con tres nodos (Linux, Windows, Cliente) que permita comunicación interna aislada mediante una red interna (red_sistemas), manteniendo acceso a internet para gestión de paquetes, y preparar los sistemas para futuras automatizaciones.

Diagrama de Topología



Guía de Uso de los Scripts

Requisitos Previos

- Los scripts deben ejecutarse en el sistema operativo correspondiente.
- En Linux: permisos de ejecución (`chmod +x`).
- En Windows: política de ejecución de PowerShell habilitada (`Set-ExecutionPolicy RemoteSigned`).

Instrucciones de Uso

Linux Arch/Ubuntu:

```
chmod +x check_status.sh  
./check_status.sh
```

Windows (Server 2022):

```
Set-ExecutionPolicy RemoteSigned -Scope  
CurrentUser  
.\\check_status.ps1
```

Flujo de Interacción

El script no solicita entrada; muestra automáticamente:

1. Nombre del equipo
2. Direcciones IP activas
3. Espacio disponible en disco principal

Bitácora de Desarrollo y Configuración

Explicación del Script

`check_status.sh` (Linux):

- Usa `cat /etc/hostname` para obtener el nombre del equipo.
- Utiliza `ip -brief addr show` para listar interfaces e IPs.
- Emplea `df -h` para mostrar espacio en disco con formato legible.

`check_status.ps1` (Windows):

- Accede a `$env:COMPUTERNAME` para el nombre del equipo.
- Usa `Get-NetIPAddress` para listar IPs IPv4.
- Utiliza `Get-PSDrive C` para obtener espacio en disco en GB.

Evidencias de Configuración

- Captura del archivo /etc/hostname en Arch:

```
[alda@archlinux ~]$ cat /etc/hostname  
Srv-Linux-Sistemas  
[alda@archlinux ~]$ _
```

- Captura de la configuración de red en Windows (`ipconfig /all`):

```
C:\> Administrador: C:\Windows\system32\cmd.exe
DUID de cliente DHCPv6. . . . . : 00-01-00-01-31-16-46-22-08-00-27-75-F6-D4
Servidores DNS. . . . . : 200.52.162.137
                           200.52.167.3
                           200.52.160.200
NetBIOS sobre TCP/IP. . . . . : habilitado

Adaptador de Ethernet Ethernet 2:

Sufijo DNS específico para la conexión. . . :
Descripción . . . . . : Intel(R) PRO/1000 MT Desktop Adapter #2
Dirección física. . . . . : 08-00-27-2D-D7-34
DHCP habilitado . . . . . : no
Configuración automática habilitada . . . : sí
Vínculo: dirección IPv6 local. . . . : fe80::1d7f:5656:d0ca:40c%6(Preferido)
Dirección IPv4. . . . . : 192.168.10.20(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . : 192.168.10.1
IAID DHCPv6 . . . . . : 168296487
DUID de cliente DHCPv6. . . . . : 00-01-00-01-31-16-46-22-08-00-27-75-F6-D4
Servidores DNS. . . . . : <cursor>fec0::0:ffff::1%1
                           fec0::0:ffff::2%1
                           fec0::0:ffff::3%1
NetBIOS sobre TCP/IP. . . . . : habilitado
```

- Captura del archivo de Netplan en Ubuntu (`/etc/netplan/01-netcfg.yaml`):

```
cli-sistema@cli-sistema-VirtualBox:/mnt$ sudo cat /etc/netplan/01-netcfg.yaml
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true    # NAT
    enp0s8:
      addresses:
        - 192.168.10.30/24
      routes:
        - to: default
          via: 192.168.10.1
      nameservers:
        addresses: [8.8.8.8]
```

Pruebas de Funcionamiento

Tabla de Direccionamiento IP

| Nodo | Sistema | IP en 'red_sistemas' |
|--------------------|---------------------|----------------------|
| Srv-Linux-Sistemas | Arch Linux | 192.168.10.10 |
| Srv-Win-Sistemas | Windows Server 2022 | 192.168.10.20 |
| Cli-Sistema | Ubuntu Linux | 192.168.10.30 |

Protocolo de Pruebas - Ping

| Origen | Destino | Esperado | Obtenido |
|---------------|---------------|----------------|----------|
| 192.168.10.10 | 192.168.10.20 | Respuesta ICMP | Si |
| 192.168.10.10 | 192.168.10.30 | Respuesta ICMP | Si |
| 192.168.10.20 | 192.168.10.10 | Respuesta ICMP | Si |
| 192.168.10.20 | 192.168.10.30 | Respuesta ICMP | Si |
| 192.168.10.30 | 192.168.10.10 | Respuesta ICMP | Si |
| 192.168.10.30 | 192.168.10.20 | Respuesta ICMP | Si |

Capturas de Validación

Captura de ping exitoso entre los tres nodos:

- 192.168.10.10 -> 192.168.10.20:

```
[alda@archlinux ~]$ ping 192.168.10.20
PING 192.168.10.20 (192.168.10.20) 56(84) bytes de datos.
64 bytes desde 192.168.10.20: icmp_seq=1 ttl=128 tiempo=1.02 ms
64 bytes desde 192.168.10.20: icmp_seq=2 ttl=128 tiempo=1.71 ms
64 bytes desde 192.168.10.20: icmp_seq=3 ttl=128 tiempo=1.65 ms
64 bytes desde 192.168.10.20: icmp_seq=4 ttl=128 tiempo=1.25 ms
64 bytes desde 192.168.10.20: icmp_seq=5 ttl=128 tiempo=0.691 ms
^C
--- 192.168.10.20 estadísticas ping ---
5 paquetes transmitidos, 5 recibidos, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.691/1.264/1.709/0.383 ms
[alda@archlinux ~]$
```

- 192.168.10.10 -> 192.168.10.30:

```
[alda@archlinux ~]$ ping 192.168.10.30
PING 192.168.10.30 (192.168.10.30) 56(84) bytes de datos.
64 bytes desde 192.168.10.30: icmp_seq=1 ttl=64 tiempo=0.987 ms
64 bytes desde 192.168.10.30: icmp_seq=2 ttl=64 tiempo=1.58 ms
64 bytes desde 192.168.10.30: icmp_seq=3 ttl=64 tiempo=1.31 ms
64 bytes desde 192.168.10.30: icmp_seq=4 ttl=64 tiempo=1.13 ms
64 bytes desde 192.168.10.30: icmp_seq=5 ttl=64 tiempo=1.39 ms
^C
--- 192.168.10.30 estadísticas ping ---
5 paquetes transmitidos, 5 recibidos, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.987/1.278/1.579/0.206 ms
```

- 192.168.10.20 -> 192.168.10.10:

```
ca Administrador: C:\Windows\system32\cmd.exe
PS C:\> ping 192.168.10.10

Haciendo ping a 192.168.10.10 con 32 bytes de datos:
Respuesta desde 192.168.10.10: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.10: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.10.10: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.10.10: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 192.168.10.10:
    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:\>
```

- 192.168.10.20 -> 192.168.10.30:

```
Administrator: C:\Windows\system32\cmd.exe
PS C:\> ping 192.168.10.30

Haciendo ping a 192.168.10.30 con 32 bytes de datos:
Respuesta desde 192.168.10.30: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.30: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.30: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.10.30: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 192.168.10.30:
    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:\>
```

- 192.168.10.30 -> 192.168.10.10:

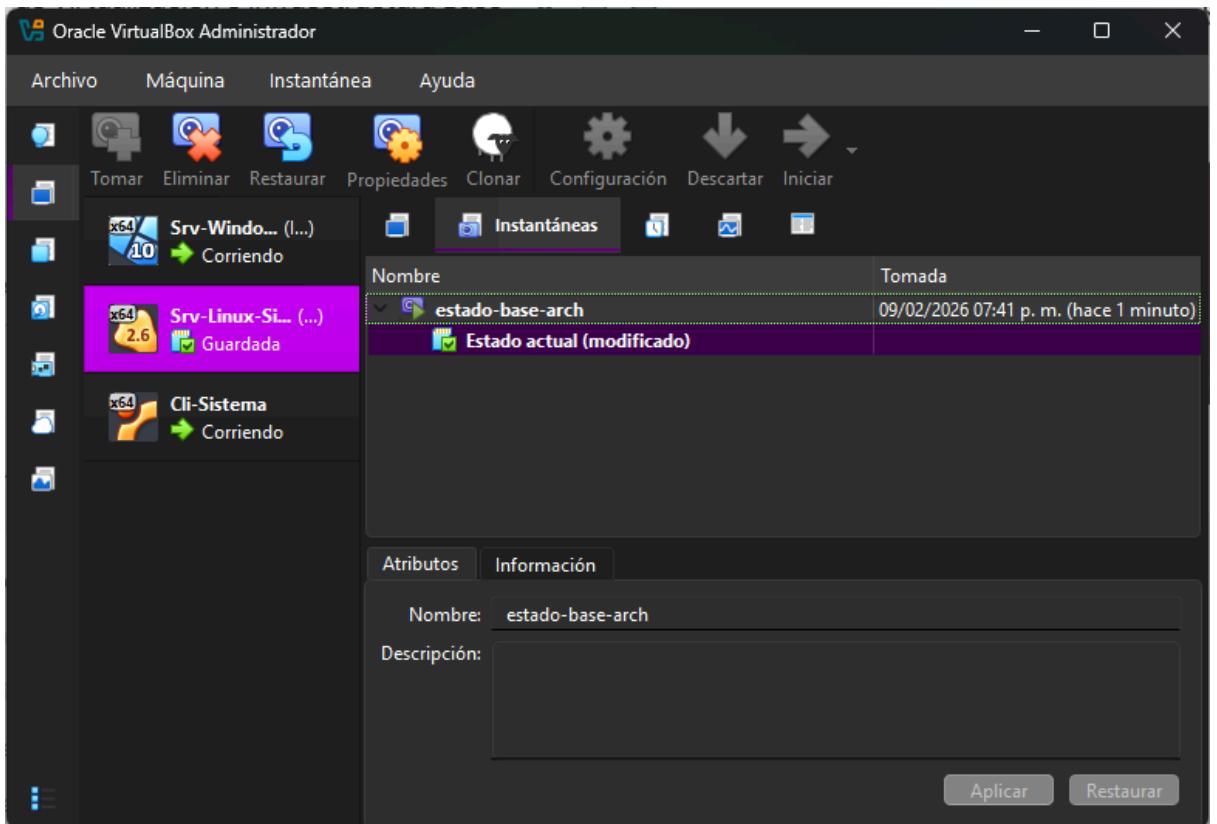
```
cli-sistema@cli-sistema-VirtualBox:/mnt$ ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=64 time=0.469 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=64 time=1.45 ms
64 bytes from 192.168.10.10: icmp_seq=3 ttl=64 time=0.785 ms
^C
--- 192.168.10.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.469/0.902/1.452/0.409 ms
```

- 192.168.10.30 -> 192.168.10.20:

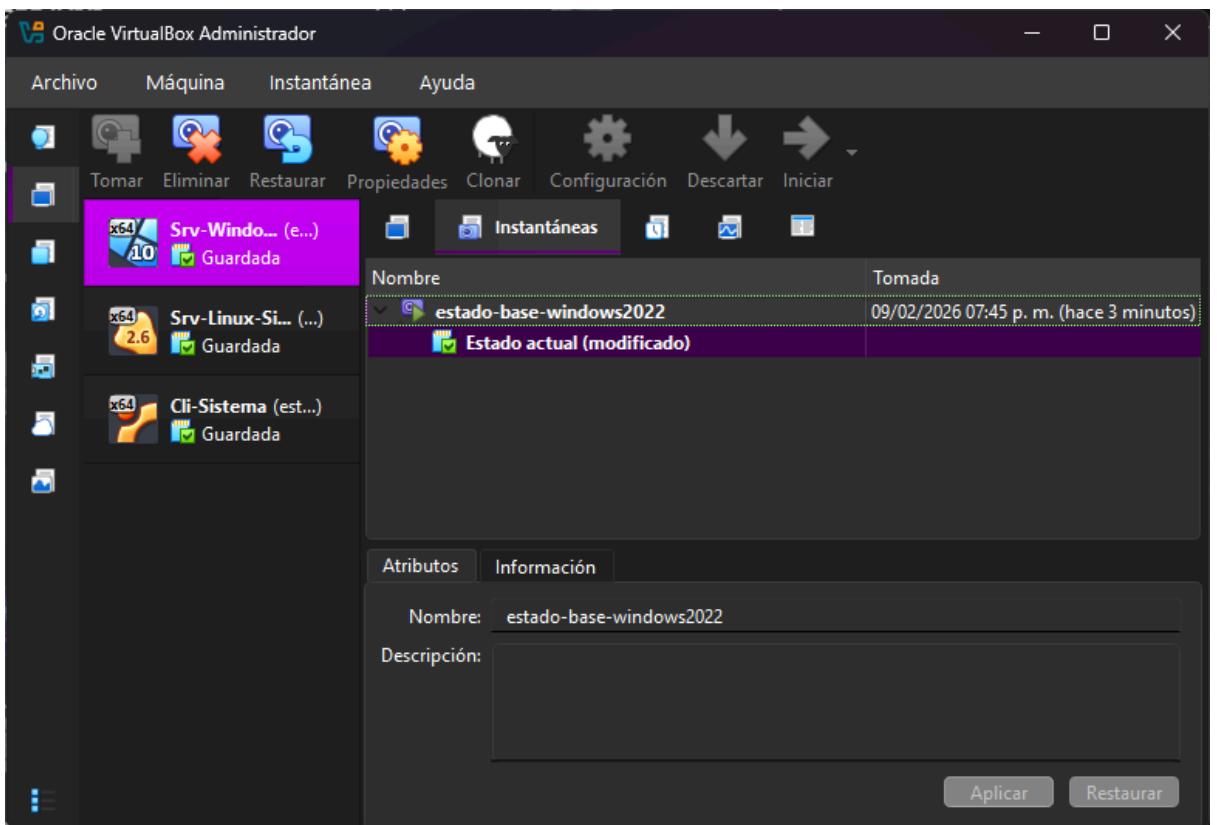
```
cli-sistema@cli-sistema-VirtualBox:/mnt$ ping 192.168.10.20
PING 192.168.10.20 (192.168.10.20) 56(84) bytes of data.
64 bytes from 192.168.10.20: icmp_seq=1 ttl=128 time=0.455 ms
64 bytes from 192.168.10.20: icmp_seq=2 ttl=128 time=1.51 ms
64 bytes from 192.168.10.20: icmp_seq=3 ttl=128 time=1.39 ms
^C
--- 192.168.10.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 0.455/1.115/1.505/0.469 ms
```

Captura de los snapshots en VirtualBox:

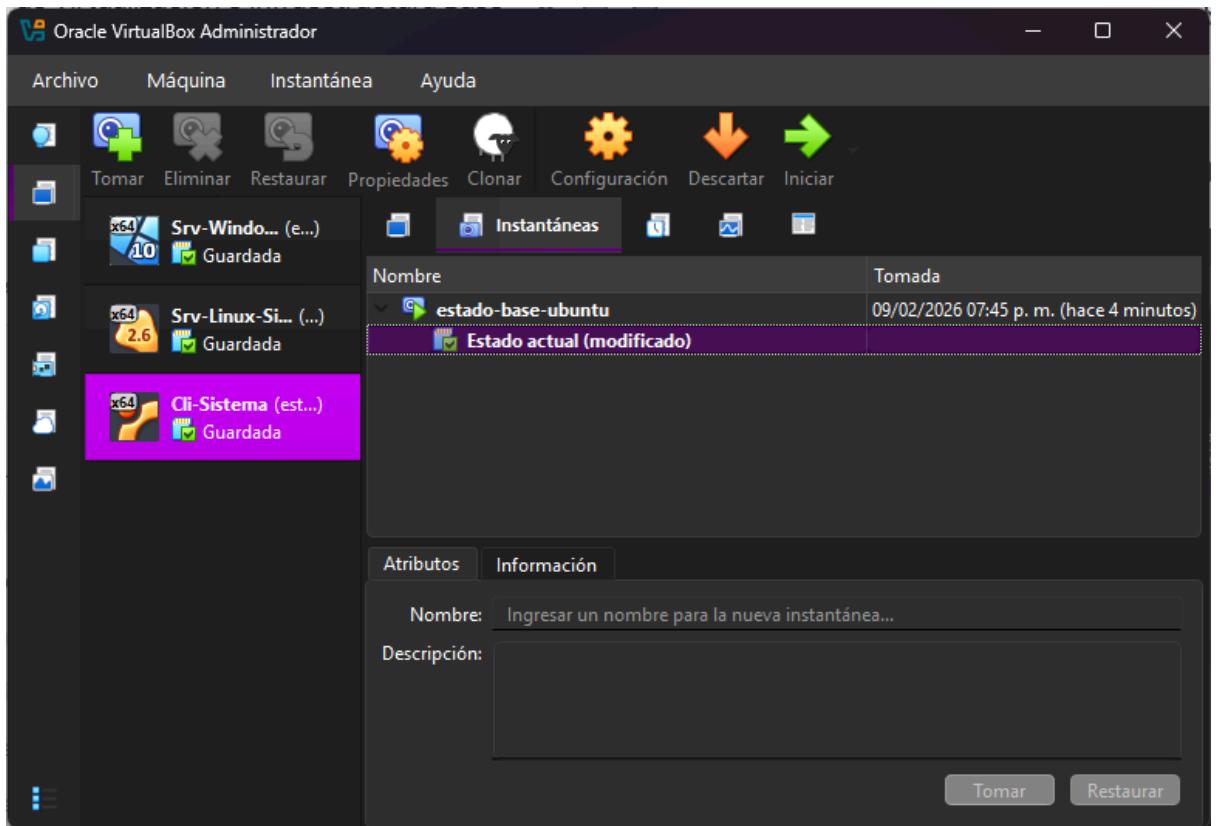
- Estado Base (Arch):



- Estado Base (Windows Server 2022):



- Estado Base (Ubuntu):



Conclusiones y Referencias

Lecciones Aprendidas

- La configuración de red interna en VirtualBox requiere que el nombre de la red sea idéntico en todas las máquinas.
- En Windows Server, el firewall bloquea ICMP por defecto; es necesario deshabilitarlo temporalmente o crear una regla.
- Arch Linux no incluye el comando hostname por defecto en instalaciones mínimas; se debe usar cat /etc/hostname.
- Netplan en Ubuntu es sensible a la indentación en YAML; un espacio de más puede romper la configuración.

Bibliografía

1. Arch Linux Installation Guide -
https://wiki.archlinux.org/title/Installation_guide
2. Curso de Máquinas Virtuales Desde Cero -
<https://youtu.be/uiFZUfmFAus?si=eAClrRh9K6xhAg8m>
3. Asistencia IA: DeepSeek Chat – <https://chat.deepseek.com>