

# TRABAJO PRÁCTICO - VIRTUALIZACIÓN

## Arquitectura y Sistemas Operativos

Profesor: Martin Aristiaran

Alumnos:

- Pascutti, Valentina - [valenpascutti2612@gmail.com](mailto:valenpascutti2612@gmail.com)
- Ramos, Angela Irina - [ramosangelairina@gmail.com](mailto:ramosangelairina@gmail.com)

FECHA ENTREGA

05/06/2025

## **ÍNDICE**

1. Introducción
2. Marco teórico
3. Caso práctico
4. Capturas de pantallas
5. Metodología utilizada
6. Resultados obtenidos
7. Desafíos superados
8. Conclusión
9. Bibliografía

## **INTRODUCCIÓN**

El siguiente trabajo está enfocado en la virtualización, una tecnología que permite crear versiones virtuales de recursos informáticos como sistemas operativos, servidores, almacenamiento y redes. Originalmente desarrollada para maximizar la capacidad de los grandes mainframes, la virtualización ha evolucionado para convertirse en una pieza fundamental en la infraestructura tecnológica moderna, desde centros de datos hasta la nube. La elección de este tema se fundamenta en su creciente relevancia en el ámbito de las tecnologías de la información, donde se ha convertido en una herramienta esencial para optimizar el uso de recursos disponibles, reducir costos y agilizar procesos.

En el camino hacia la formación como técnico en programación, comprender el funcionamiento y las ventajas es muy importante. Esta tecnología no solo permite ejecutar múltiples máquinas virtuales en un mismo equipo físico, sino que también facilita el desarrollo, las pruebas y el despliegue de software en entornos seguros y controlados. Además, el aislamiento que proporciona contribuye a mejorar la seguridad al contener posibles fallos o amenazas dentro de entornos controlados.

Un aspecto central dentro de la virtualización es el uso de máquinas virtuales e hipervisores. Estas herramientas permiten la creación de entornos virtuales eficientes, seguros y escalables, base esencial para la gestión moderna de infraestructuras tecnológicas. Gracias a esta tecnología, es posible replicar distintos escenarios sin afectar el sistema principal, lo cual mejora la eficiencia y reduce riesgos durante el ciclo de vida del software.

La virtualización aporta múltiples beneficios que optimizan la gestión y el uso de la infraestructura tecnológica. En términos de eficiencia, permite aprovechar al máximo los recursos físicos del hardware, permitiendo un uso más eficiente del hardware disponible y minimizando el desperdicio de recursos, lo que se traduce en un mejor rendimiento general del sistema. Gracias a la flexibilidad que ofrece, los desarrolladores y administradores pueden crear entornos virtuales aislados para realizar pruebas, desarrollar nuevas aplicaciones y desplegar software de manera segura, sin afectar otros sistemas o procesos. Esto facilita la innovación y acelera los ciclos de desarrollo. Por otro lado, la virtualización simplifica la escalabilidad, ya que es posible ajustar rápidamente la cantidad de recursos asignados en función de las demandas del momento, sin necesidad de comprar o instalar hardware físico adicional, lo que también mejora la capacidad de respuesta ante cambios o picos de trabajo. Finalmente, la virtualización genera un ahorro considerable en costos al

minimizar la necesidad de servidores físicos, reducir el consumo energético y disminuir los gastos asociados al mantenimiento y espacio físico, lo que resulta en una operación mas económica y sostenible a largo plazo.

## MARCO TEÓRICO

- ☐ **Virtualización de hardware:** La virtualización es una tecnología para crear versiones virtuales, tales como sistemas operativos, servidores o almacenamiento. Se hace uso de unas instrucciones que nos permiten optimizar el funcionamiento de una o varias máquinas virtuales sobre un mismo hardware físico.
- ☐ **Hipervisores:** Un hipervisor es un programa que posibilita la creación y administración de máquinas virtuales. Su función principal es actuar como capa intermedia entre el hardware físico y los sistemas operativos de las máquinas virtuales, garantizando que cada una de ellas utilice los recursos de manera controlada y eficiente. Existen dos tipos principales de hipervisores: los de tipo 1 (Nativo o Bare-metal), que se instalan directamente sobre el hardware y ofrecen un rendimiento óptimo, comúnmente usados en entornos empresariales, requieren de una configuración más compleja y software dedicado; y los de tipo 2 (Alojado), que se ejecutan sobre un sistema operativo anfitrión, utilizando APIs y servicios del mismo para gestionar recursos, siendo más adecuados para usuarios finales, aprendizaje o entornos menos exigentes. Esta estructura permite la creación de entornos virtuales eficientes, seguros y escalables, base esencial para la gestión moderna de infraestructuras tecnológicas.
- ☐ **Máquina virtual:** Una máquina virtual es un equipo definido por software que funciona de manera independiente sobre un equipo físico, llamado máquina host, permitiendo ejecutar varios sistemas operativos y aplicaciones en un mismo hardware. Para gestionar estas máquinas virtuales, se utiliza un software especializado llamado hipervisor, que administra la asignación de recursos y asegura que cada máquina virtual opere sin interferir con las demás.
- ☐ **Redes virtuales:** Las redes virtuales permiten conectar máquinas virtuales entre sí. Estas redes no son físicas, sino que son configuradas por software dentro del programa de virtualización (como VirtualBox, VMware, etc.). Se comportan como si fueran redes reales, pero funcionan de manera aislada o controlada, lo que permite simular distintos entornos de red sin necesidad de hardware adicional.  
En este trabajo, se utilizaron principalmente dos tipos de conexión de red para las máquinas virtuales: **NAT**, que permite el acceso a internet a través del host sin exponer la máquina a la red externa, y **Bridge**, que conecta directamente la máquina virtual a la red física, asignándole una dirección IP propia como si fuera un dispositivo más de la red.
- ☐ **Imagen ISO:** Una imagen ISO es un archivo que contiene una copia exacta de un sistema operativo o software instalable. Se usa para instalar sistemas operativos en

máquinas virtuales, del mismo modo que antes se usaban discos físicos (CD/DVD). En este trabajo, se utilizó una imagen ISO de **Ubuntu Server**, un sistema operativo ligero y optimizado para tareas de servidor.

## **CASO PRÁCTICO**

### **Conexión en Red entre Máquinas Virtuales mediante Adaptador Bridge en VirtualBox.**

En este caso práctico se simula un entorno típico de desarrollo y pruebas mediante la creación de dos máquinas virtuales conectadas a través de un adaptador puente (Bridge Adapter) en VirtualBox. Este tipo de conexión permite que las máquinas virtuales se comporten como si fueran equipos físicos independientes conectados a la misma red local que el equipo anfitrión.

El objetivo principal es demostrar la comunicación entre ambas máquinas virtuales, configuradas con Ubuntu Server. Una de las VMs funciona como servidor web, con Apache instalado, mientras que la otra actúa como cliente, accediendo al contenido publicado por el servidor. Esta configuración permite que ambas VMs obtengan una dirección IP desde la red física del host y pueden ser accesibles entre sí, e incluso desde otros dispositivos de la red local. Es una forma eficaz de simular entornos reales de trabajo en red utilizando tecnología de virtualización.

#### **Objetivo:**

- Crear dos máquinas virtuales con Ubuntu Server
- VM1: Configurada como servidor web con Apache instalado
- VM2: Configurada como cliente para acceder al servidor web.
- Ambas tienen que estar conectadas a la red física del host mediante un adaptador puente

#### **Requisitos:**


- VirtualBox instalado.
- Imagen ISO de Ubuntu Server 24.04
- Conocimientos de la terminal Linux

#### **Pasos:**

- Crear dos VMs con 3 GB de RAM, 30 GB de disco y 2 procesadores.
- Cargar la imagen ISO de Ubuntu Server para instalar el sistema operativo en ambas máquinas.
- Configurar la red. Adaptador Bridge para que puedan comunicarse entre sí y con el host.
- Instalar Ubuntu Server.
- Instalar Apache en una de las VMs.  
`sudo apt update`  
`sudo apt install apache2`
- En VM2 acceder al servidor desde el navegador o curl.

## •Creación máquina virtual 1 (VM1-ServidorApache-AYSO):

Crear máquina virtual



Nombre y sistema operativo

Nombre: VM1-ServidorApache-AYSO ✓

Carpeta: C:\Users\Usuario\Documents\C...ectura y Sitemas Operativos\TP-Integrador-Virtualizacion

Imagen ISO: C:\Users\Usuario\Downloads\ubuntu-24.04.2-live-server-amd64.iso ✓

Edición:

Tipo: Linux x64

Subtype: Ubuntu

Versión: Ubuntu (64-bit)

☐ Omitir instalación desatendida


> Instalación desatendida

> Hardware

> Disco duro

Ayuda Anterior Terminar Cancelar

Crear máquina virtual



> Nombre y sistema operativo

Instalación desatendida

Usuario y contraseña

Nombre de usuario: Angela ✓

Contraseña: ●●●●

Repetir contraseña: ●●●●

Opciones adicionales

Clave de producto: #####-#####-#####-#####-#####

Nombre de máquina: VM1-ServidorApache-AYSO ✓

Nombre de dominio: myguest.virtualbox.org ✓

☐ Instalar en segundo plano

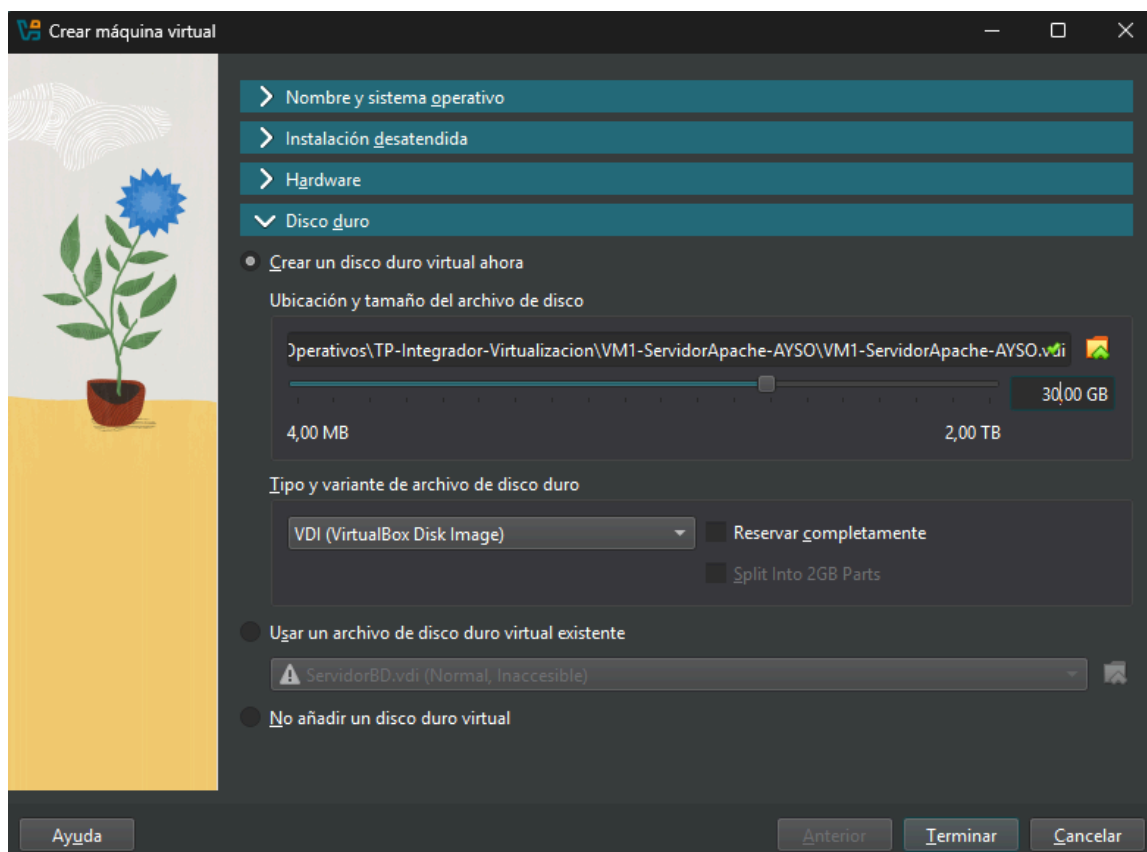
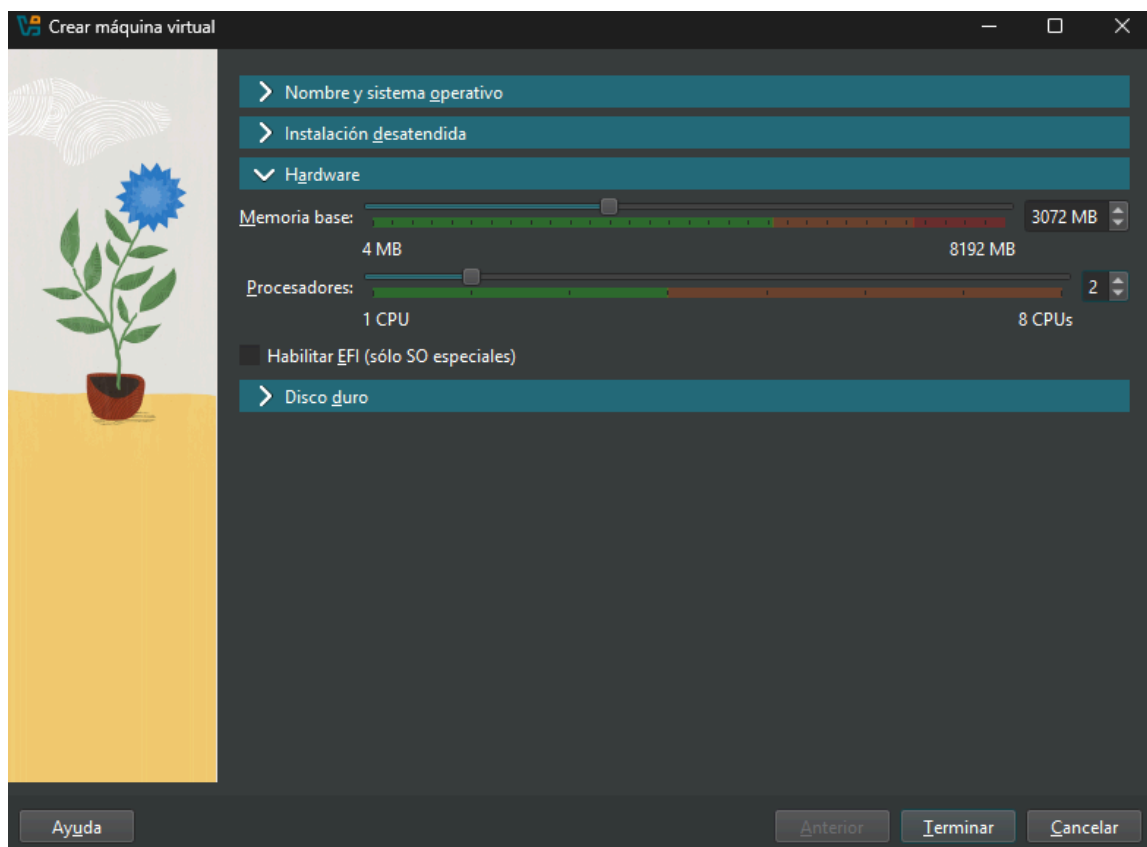
Complementos del invitado

ISO de complementos del invitado: C:\Program Files\Oracle\VirtualBox\VBBoxGuestAdditions.iso

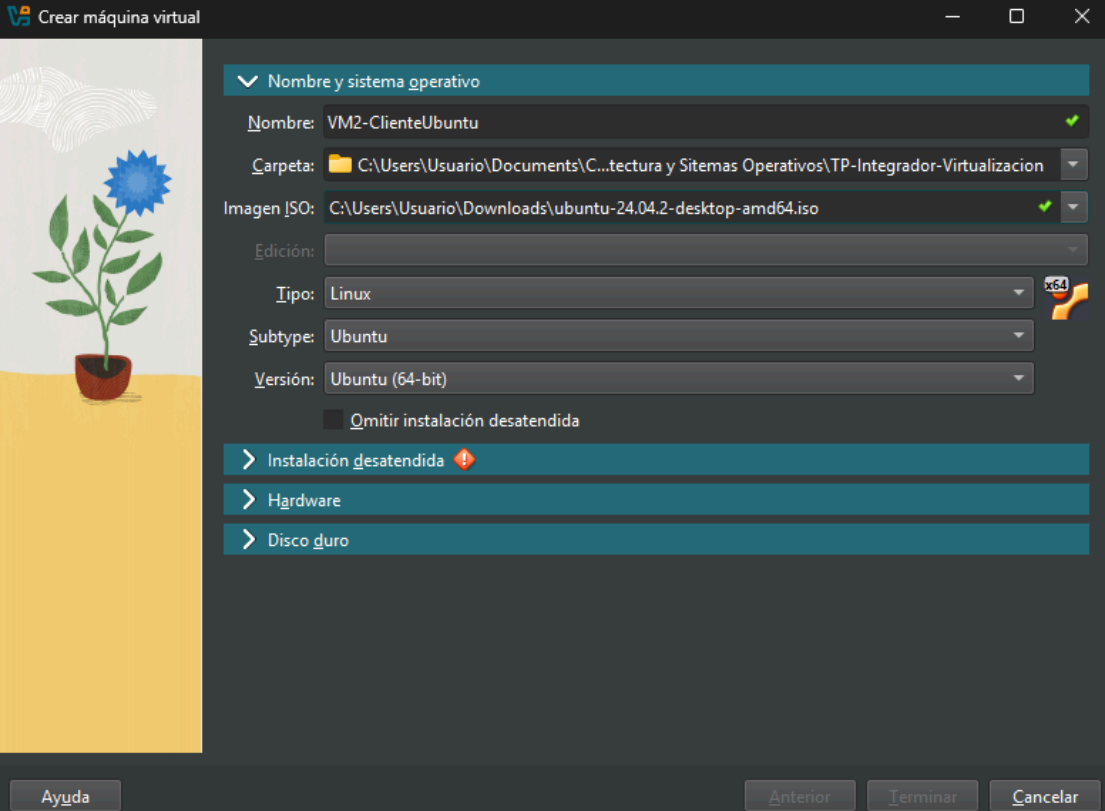
> Hardware

> Disco duro

Ayuda Anterior Terminar Cancelar



## • Creación máquina virtual 2 (VM2-ClienteUbuntu-AYSO):



Crear máquina virtual

Nombre y sistema operativo

Nombre: VM2-ClienteUbuntu ✓

Carpeta: C:\Users\Usuario\Documents\C...tectura y Sitemas Operativos\TP-Integrador-Virtualizacion

Imagen ISO: C:\Users\Usuario\Downloads\ubuntu-24.04.2-desktop-amd64.iso ✓

Edición:

Tipo: Linux x64

Subtipo: Ubuntu

Versión: Ubuntu (64-bit)

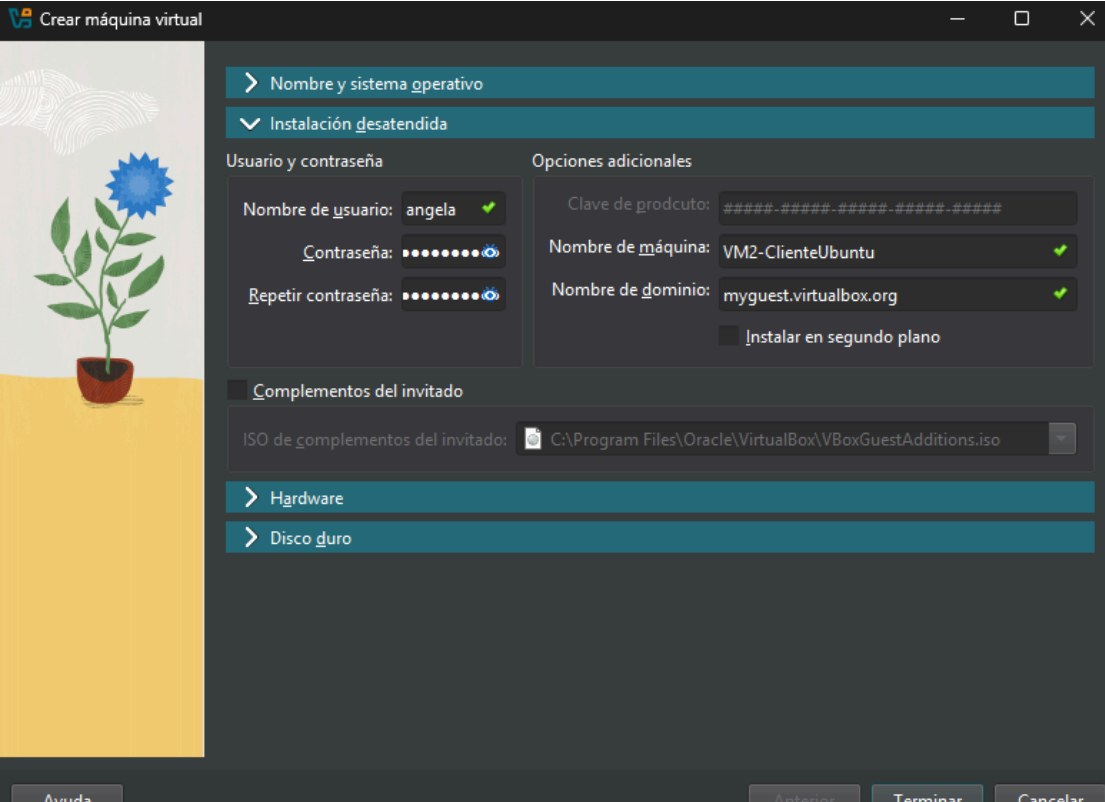
☐ Omitir instalación desatendida

> Instalación desatendida ⚠

> Hardware

> Disco duro

Ayuda Anterior Terminar Cancelar



Crear máquina virtual

> Nombre y sistema operativo

✓ Instalación desatendida

Usuario y contraseña

Nombre de usuario: angela ✓

Contraseña: ••••••••

Repetir contraseña: ••••••••

Opciones adicionales

Clave de producto: #####-####-####-####-####

Nombre de máquina: VM2-ClienteUbuntu ✓

Nombre de dominio: myguest.virtualbox.org ✓

☐ Instalar en segundo plano

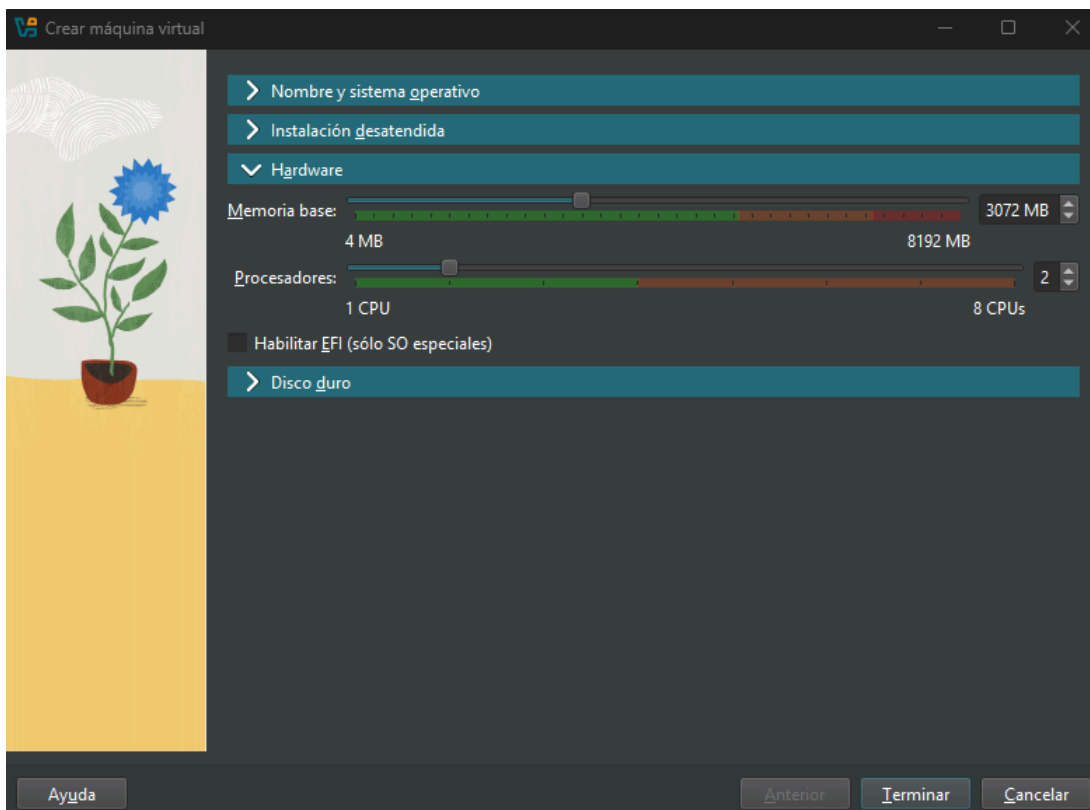
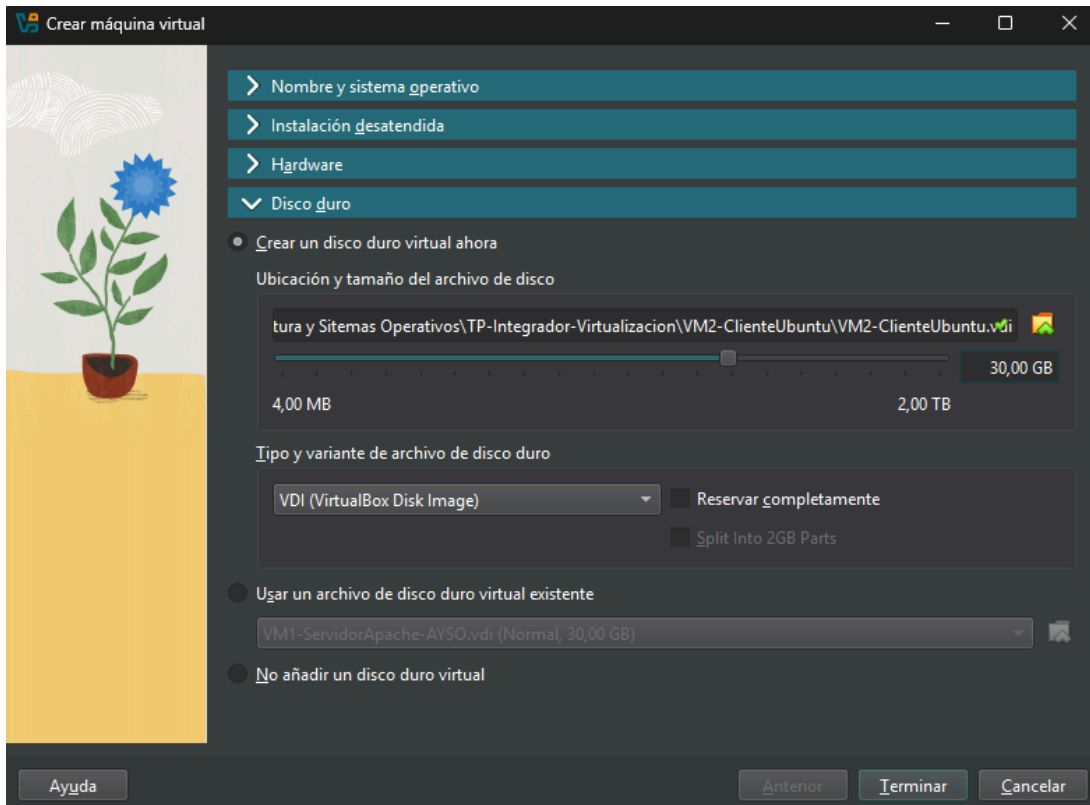
☐ Complementos del invitado

ISO de complementos del invitado: C:\Program Files\Oracle\VirtualBox\VBBoxGuestAdditions.iso

> Hardware

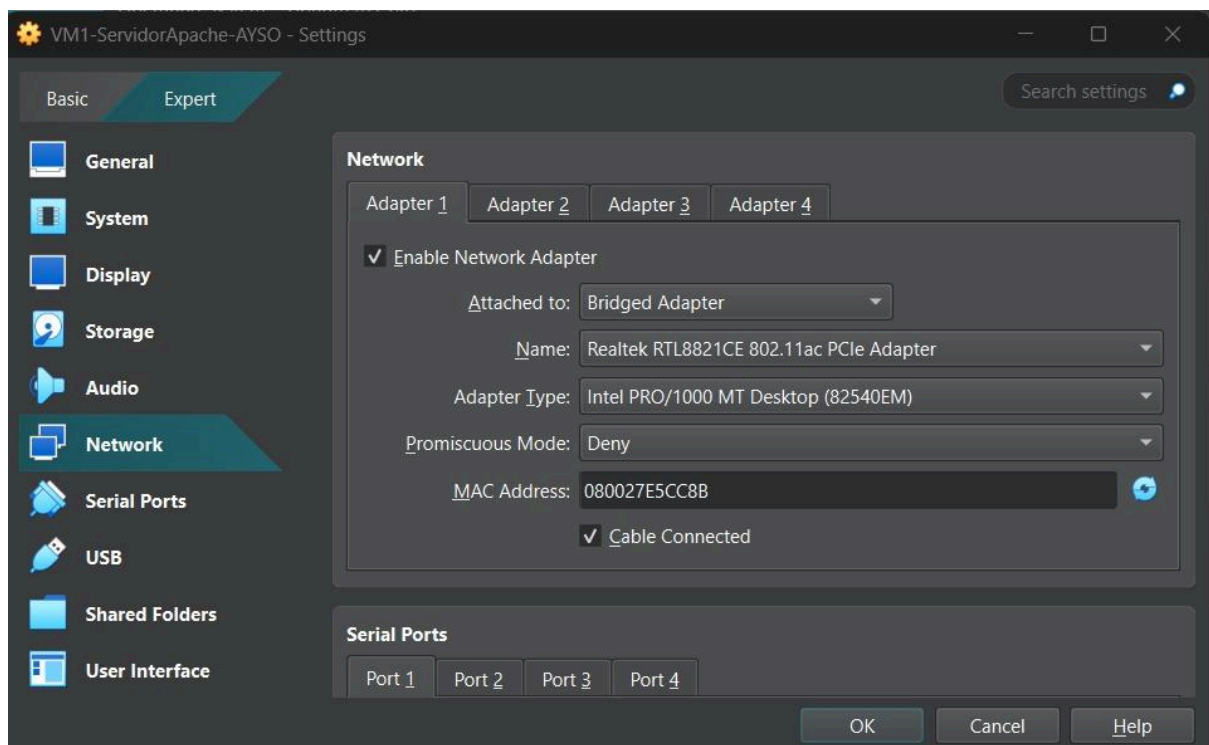
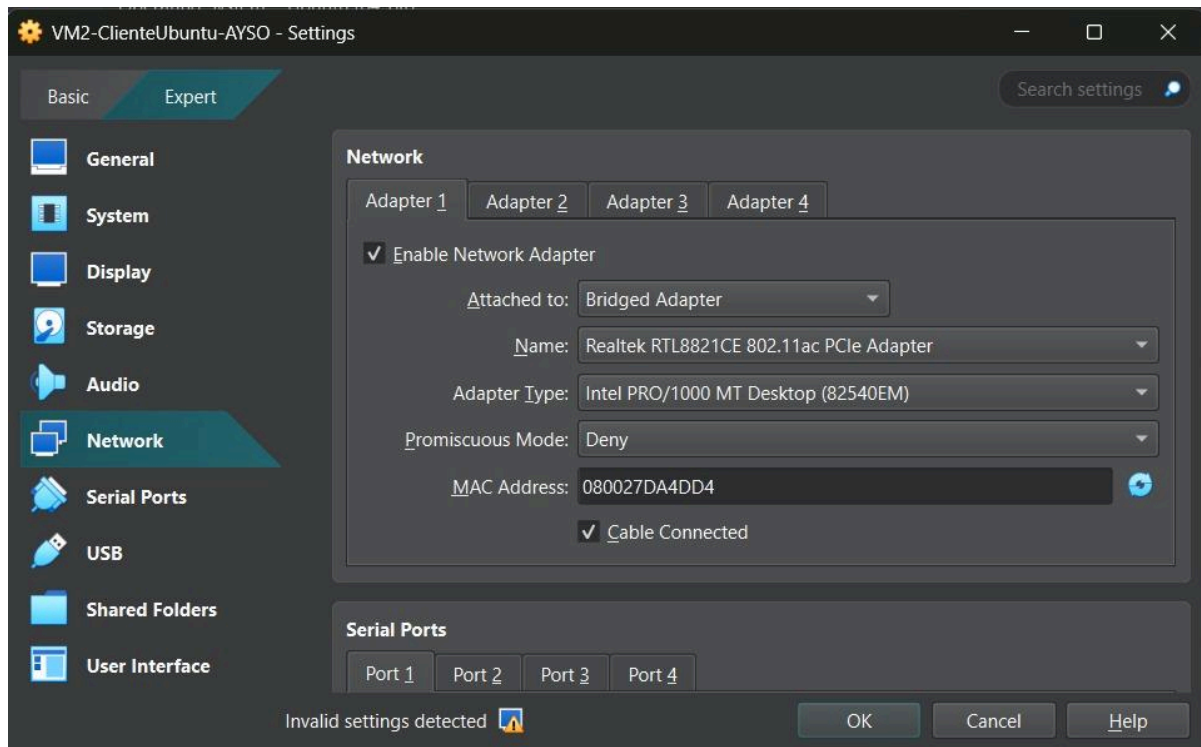
> Disco duro

Ayuda Anterior Terminar Cancelar



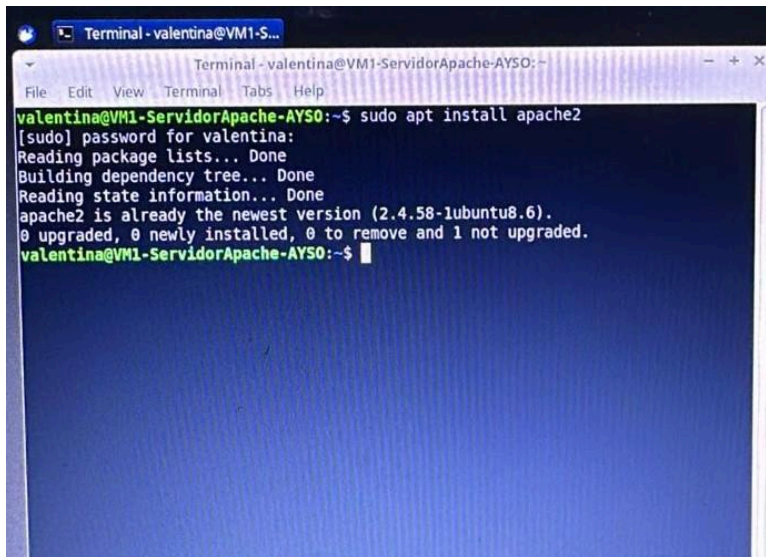


## •Configuración de la red



## •Instalación del Servidor Apache

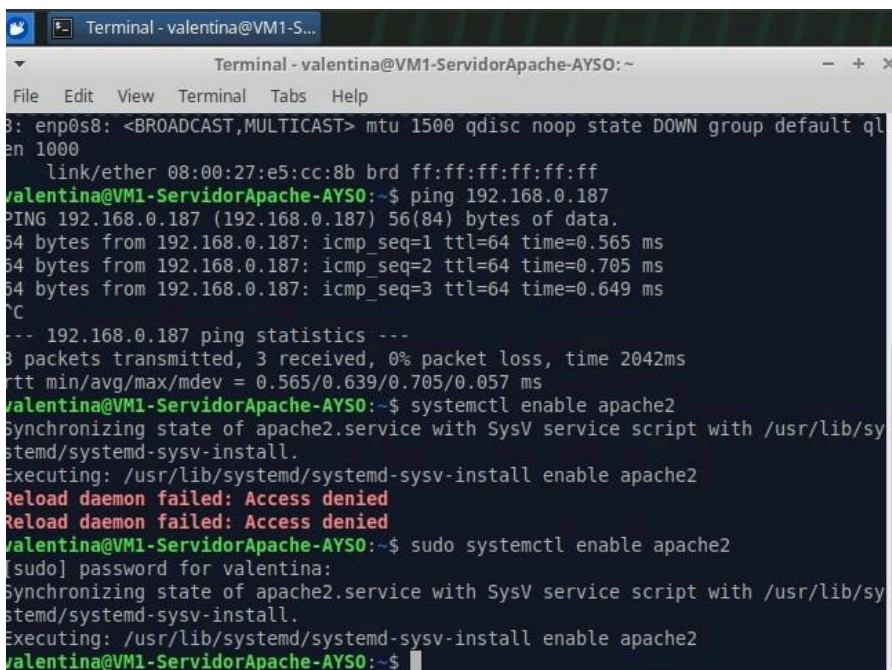
Se ejecuta el comando `sudo apt install apache2` para instalar el servidor web Apache en una máquina con sistema operativo basado en Ubuntu. El mensaje indica que Apache ya está instalado y se encuentra en su versión más reciente (2.4.58).



```
Terminal - valentina@VM1-S...
Terminal - valentina@VM1-ServidorApache-AYSO: ~
valentina@VM1-ServidorApache-AYSO:~$ sudo apt install apache2
[sudo] password for valentina:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.6).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
valentina@VM1-ServidorApache-AYSO:~$
```

## •Habilitación del servicio Apache y prueba de conectividad

Se realiza una prueba de red con `ping` para verificar la conexión con otra máquina (IP 192.168.0.187). Luego, se intenta habilitar el servicio Apache con `systemctl enable apache2`, pero inicialmente falla por falta de permisos. Al corregirse el comando con `sudo`, se habilita correctamente el servicio.



```
Terminal - valentina@VM1-S...
Terminal - valentina@VM1-ServidorApache-AYSO: ~
8: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default ql
en 1000
    link/ether 08:00:27:e5:cc:8b brd ff:ff:ff:ff:ff:ff
valentina@VM1-ServidorApache-AYSO:~$ ping 192.168.0.187
PING 192.168.0.187 (192.168.0.187) 56(84) bytes of data.
64 bytes from 192.168.0.187: icmp_seq=1 ttl=64 time=0.565 ms
64 bytes from 192.168.0.187: icmp_seq=2 ttl=64 time=0.705 ms
64 bytes from 192.168.0.187: icmp_seq=3 ttl=64 time=0.649 ms
^C
--- 192.168.0.187 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.565/0.639/0.705/0.057 ms
valentina@VM1-ServidorApache-AYSO:~$ systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/sy
stemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
Reload daemon failed: Access denied
Reload daemon failed: Access denied
valentina@VM1-ServidorApache-AYSO:~$ sudo systemctl enable apache2
[sudo] password for valentina:
Synchronizing state of apache2.service with SysV service script with /usr/lib/sy
stemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
valentina@VM1-ServidorApache-AYSO:~$
```

## •Verificación de dirección IP del servidor

Se utiliza el comando `ip a` para visualizar las interfaces de red y sus direcciones IP. En este caso, se identifica que la interfaz `enp0s3` tiene la IP `10.0.2.15`, la cual es útil para acceder al servidor Apache desde otro dispositivo dentro de la misma red.

```
valentina@VM1-ServidorApache-AYS0:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid lft forever preferred lft forever
    inet6 ::1/128 scope host noprefixroute
        valid lft forever preferred lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:36:28:78 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid lft 86034sec preferred lft 86034sec
    inet6 fd17:625c:f037:2:a00:27ff:fe36:2878/64 scope global dynamic mngtmpaddr noprefixroute
        valid lft 86036sec preferred lft 14036sec
    inet6 fe80::a00:27ff:fe36:2878/64 scope link
        valid lft forever preferred lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:03:dc:4f brd ff:ff:ff:ff:ff:ff
valentina@VM1-ServidorApache-AYS0:~$
```

## •Verificación de Puertos en Escucha en el Servidor

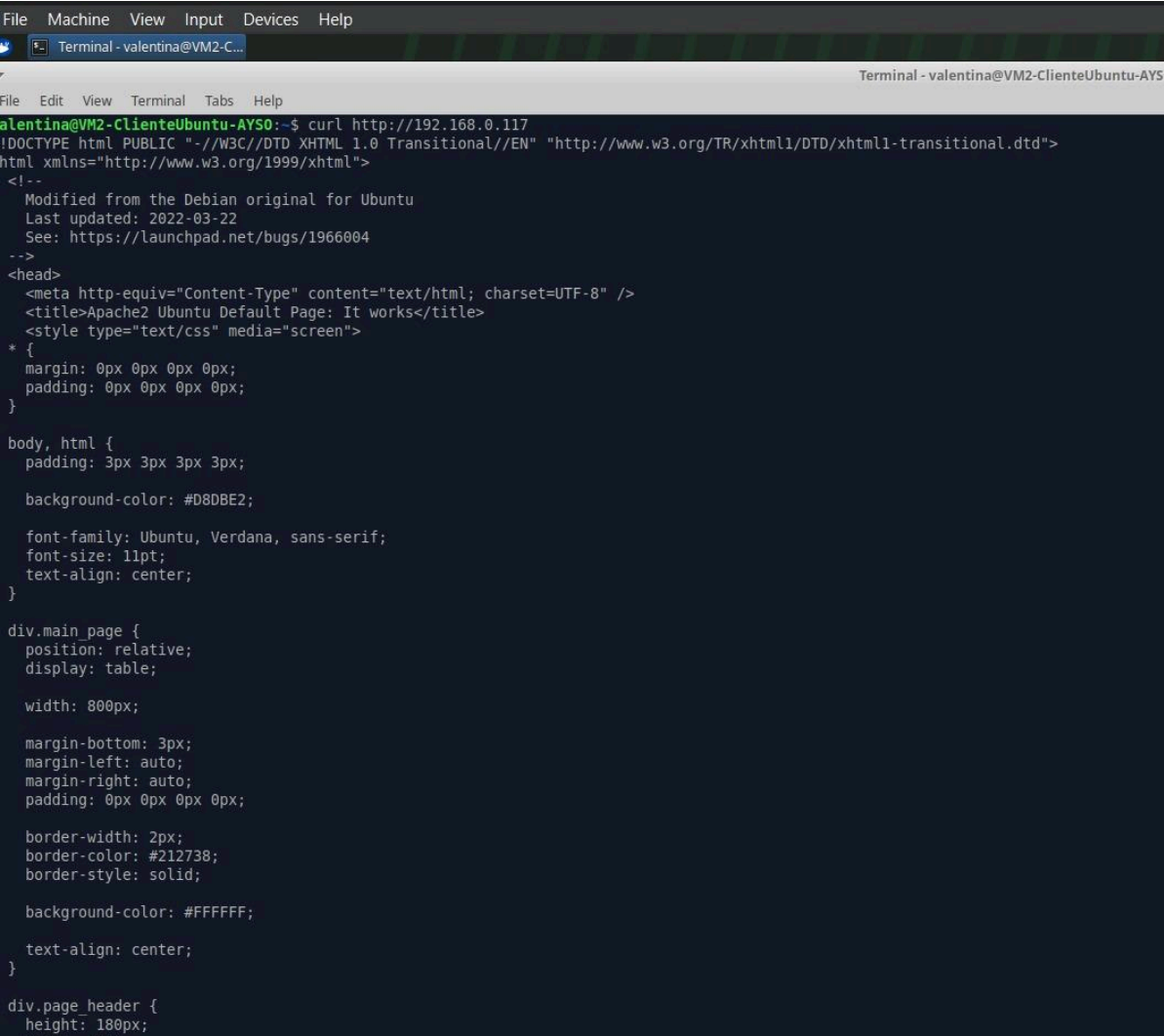
Desde la máquina servidor (VM1), se utiliza el comando `sudo lsof -i -P -n | grep LISTEN` para listar los procesos que están escuchando en puertos de red. Se confirma que el servicio `apache2` está escuchando en el puerto 80 (HTTP), lo cual valida que el servidor web está activo y disponible para recibir conexiones entrantes.

```
Lines 1-16/16 (END)
valentina@VM1-ServidorApache-AYS0:~$ http
valentina@VM1-ServidorApache-AYS0:~$ sudo isof -i -p -n | grep LISTEN
sudo: isof: command not found
valentina@VM1-ServidorApache-AYS0:~$ sudo lsof -i -p -n | grep LISTEN
lsof: illegal process ID: -n
lsof 4.95.0
latest revision: https://github.com/lsof-org/lsof
latest FAQ: https://github.com/lsof-org/lsof/blob/master/00FAQ
latest (non-formatted) man page: https://github.com/lsof-org/lsof/blob/master/Lsof.8
usage: [-?abhKlnNoOPRtUvVX] [+|-c c] [+|-d s] [+D D] [+|-E] [+|-e s] [+|-f[gG]]
[-F [f]] [-g [s]] [-i [i]] [+|-L [l]] [+m [m]] [+|-M] [-o [o]] [-p s]
[+|-r [t]] [-s [p:s]] [-S [t]] [-T [t]] [-u s] [+|-w] [-x [f]] [--] [names]
Use the ``-h'' option to get more help information.
valentina@VM1-ServidorApache-AYS0:~$ sudo lsof -i -P -n | grep LISTEN
systemd-r 430 systemd-resolve 15u IPv4 6434 0t0 TCP 127.0.0.53:53 (LISTEN)
systemd-r 430 systemd-resolve 17u IPv4 6436 0t0 TCP 127.0.0.54:53 (LISTEN)
cupsd 1025 root 6u IPv6 9495 0t0 TCP [::]:631 (LISTEN)
cupsd 1025 root 7u IPv4 9496 0t0 TCP 127.0.0.1:631 (LISTEN)
apache2 1091 root 3u IPv6 10036 0t0 TCP *:80 (LISTEN)
apache2 1092 www-data 3u IPv6 10036 0t0 TCP *:80 (LISTEN)
apache2 1094 www-data 3u IPv6 10036 0t0 TCP *:80 (LISTEN)
valentina@VM1-ServidorApache-AYS0:~$
```



## •Verificación del Servidor Apache desde el Cliente

Desde la máquina cliente (VM2), se ejecuta el comando `curl http://192.168.0.117` para comprobar que el servidor Apache está funcionando correctamente en la VM del servidor (VM1). La salida muestra el HTML por defecto de Apache, lo cual indica que la conexión entre cliente y servidor es exitosa y que Apache está sirviendo páginas correctamente.



```
File Machine View Input Devices Help
Terminal - valentina@VM2-C...

File Edit View Terminal Tabs Help
Terminal - valentina@VM2-ClienteUbuntu-AYS...

valentina@VM2-ClienteUbuntu-AYS0:~$ curl http://192.168.0.117
!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
html xmlns="http://www.w3.org/1999/xhtml">
<!--
  Modified from the Debian original for Ubuntu
  Last updated: 2022-03-22
  See: https://launchpad.net/bugs/1966004
-->
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Apache2 Ubuntu Default Page: It works</title>
  <style type="text/css" media="screen">
* {
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}

body, html {
  padding: 3px 3px 3px 3px;

  background-color: #D8DBE2;

  font-family: Ubuntu, Verdana, sans-serif;
  font-size: 11pt;
  text-align: center;
}

div.main page {
  position: relative;
  display: table;

  width: 800px;

  margin-bottom: 3px;
  margin-left: auto;
  margin-right: auto;
  padding: 0px 0px 0px 0px;

  border-width: 2px;
  border-color: #212738;
  border-style: solid;

  background-color: #FFFFFFF;

  text-align: center;
}

div.page_header {
  height: 180px;
  width: 100%;
```

## •Verificación de conectividad entre host y máquinas virtuales

Se muestra la ejecución del comando **ping** desde el equipo host hacia las IP asignadas a las VMs. Ambas responden correctamente, lo que demuestra que las VMs están integradas a la red local del host, con conectividad activa y estable.

```
Command Prompt
> ipconfig /allcompartments /all ... Show detailed information about all compartments

C:\Users\Usuario>ping 192.168.0.117

Pinging 192.168.0.117 with 32 bytes of data:
Reply from 192.168.0.117: bytes=32 time=1ms TTL=64
Reply from 192.168.0.117: bytes=32 time<1ms TTL=64
Reply from 192.168.0.117: bytes=32 time=1ms TTL=64
Reply from 192.168.0.117: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.117:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Usuario>ping 192.168.0.187

Pinging 192.168.0.187 with 32 bytes of data:
Reply from 192.168.0.187: bytes=32 time=1ms TTL=64
Reply from 192.168.0.187: bytes=32 time<1ms TTL=64
Reply from 192.168.0.187: bytes=32 time<1ms TTL=64
Reply from 192.168.0.187: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.187:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

## **METODOLOGÍA UTILIZADA**

### **1. Investigación Previa**

Para comenzar el Trabajo Integrador, hicimos una investigación teórica sobre el tema elegido: virtualización. Enfocándonos en comprender los fundamentos, los distintos tipos de hipervisores, como funcionan las máquinas virtuales y las configuraciones de red asociadas. Para esto utilizamos la información brindada a través del aula virtual por la cátedra, incluyendo documentos, ejercicios y videos.

### **2. Etapas de Diseño y Prueba del Entorno Virtual**

- Selección del software de virtualización: Elegimos Oracle VirtualBox debido a su facilidad de uso y compatibilidad multiplataforma, lo que simplificó tanto el trabajo individual como el colaborativo.
- Definición de las máquinas virtuales: Planificamos la creación de dos máquinas virtuales específicas para simular los escenarios deseados.
- Configuración de red: Configuramos la red utilizando el Bridge Adapter. Esto permitió integrar directamente las máquinas virtuales en la red física del equipo anfitrión (host), asignándoles direcciones IP propias como si fueran dispositivos adicionales en la red.
- Pruebas incrementales: Finalmente, realizamos pruebas de forma progresiva. Cada componente del entorno, desde la instalación del sistema operativo hasta la configuración de red y la funcionalidad de los servicios, fue probado individualmente para asegurar su correcto funcionamiento. Esto incluyó pruebas de conectividad y de la operatividad de los servicios implementados.

### **3. Herramientas y Recursos utilizados**

- Entorno de virtualización: Oracle VirtualBox.
- Sistemas Operativos: Ubuntu Server (imagen ISO).
- Consola de VirtualBox, para la configuración inicial y monitoreo de las VMs.
- Google Docs, para la elaboración del Trabajo Integrador de manera colaborativa.

### **4. Trabajo Colaborativo**

El proyecto lo llevamos a cabo de manera colaborativa, distribuyendo las tareas para poder optimizar el tiempo. La búsqueda de información, la organización del contenido y la redacción general del informe fueron divididas equitativamente entre ambas alumnas, promoviendo una participación activa y equilibrada. El caso práctico de configuración de las máquinas virtuales se realizó de forma conjunta mediante videollamadas, lo que permitió trabajar en tiempo real, resolver dudas de manera inmediata, tomar decisiones consensuadas y mantener coherencia en cada etapa del proceso. Además, este trabajo fomenta el aprendizaje mutuo y fortalece la comunicación, aspectos clave en el trabajo en equipo dentro de entornos tecnológicos.

## **RESULTADOS OBTENIDOS**

El caso práctico permitió configurar con éxito un entorno virtual funcional, cumpliendo los objetivos planteados al inicio.

### **Exitos:**

- Implementación de Máquinas Virtuales: Se logró la creación y puesta en marcha de dos máquinas virtuales con Ubuntu Server, configuradas correctamente y operativas.
- Conectividad de Red eficiente: al usar el adaptador en modo Bridge, cada máquina virtual obtuvo una dirección IP directamente de la red física. Esto garantizó una comunicación estable entre ambas VMs, así como acceso a Internet.

## **DESAFÍOS SUPERADOS**

Durante la configuración inicial del entorno, enfrentamos un problema importante relacionado con la conectividad de la red. Al intentar comunicar ambas máquinas virtuales (VMs) utilizando un adaptador de Red Interna, surgieron dos inconvenientes principales: la pérdida de conexión a internet, lo que dificultaba la instalación de paquetes y la actualización del sistema y conflictos de red, debido a la asignación incorrecta o duplicada de direcciones IP entre las VMs.

Ambos problemas fueron resueltos de forma efectiva al modificar la configuración de red y utilizar el Adaptador en Modo Puente (Bridge Adapter). Esta solución no sólo restauró la conexión a internet para las VMs, sino que también aseguró una comunicación sin conflictos y el acceso completo a los recursos externos. Este ajuste fue clave para alcanzar el funcionamiento esperado del entorno virtualizado y continuar con el desarrollo del caso práctico sin mayores inconvenientes.

## CONCLUSIÓN

A lo largo de este trabajo pudimos ver cómo la virtualización se ha convertido en una herramienta clave tanto en el ámbito educativo como profesional. Gracias a ella, es posible crear entornos de prueba, experimentación o aprendizaje sin necesidad de contar con múltiples equipos físicos. Esto no solo reduce costos, sino que también facilita el acceso a tecnologías que de otro modo serían más difíciles de implementar o probar.

Utilizando VirtualBox, herramienta que nos permitió simular distintos sistemas operativos, entender su funcionamiento y realizar pruebas sin comprometer el sistema principal. Además, el proceso de virtualización nos ayudó a desarrollar habilidades técnicas y a comprender mejor cómo se estructura y administra un entorno informático completo.

Sin embargo, también nos encontramos con ciertos desafíos. Por ejemplo, al principio puede ser un poco complejo entender cómo asignar los recursos del sistema (como memoria o espacio en disco) de forma equilibrada para que tanto el equipo anfitrión como la máquina virtual funcionen correctamente. Además, algunos errores o incompatibilidades pueden surgir dependiendo del hardware disponible.

En resumen, este trabajo nos permitió comprobar que la virtualización no solo es una solución práctica, sino también una herramienta educativa muy valiosa. Nos dio la posibilidad de explorar y aprender en un entorno controlado, sin correr riesgos. A pesar de los pequeños desafíos técnicos, la experiencia fue enriquecedora y nos deja mejor preparados para enfrentar situaciones similares en el futuro, con una base sólida y aplicable a distintos contextos del mundo real.



## **Bibliografía:**

**Apache Software Foundation. (s.f.).** *Apache HTTP Server Version 2.4 Documentation.*  
<https://httpd.apache.org/docs/2.4/es/>

**Canonical Ltd. (s.f.).** *Ubuntu Server Documentation.*  
<https://documentation.ubuntu.com/server/>

**Enferrel, A. (s.f.).** *Virtualización: Conceptos y beneficios.* Apuntes de clase del curso Arquitectura y Sistemas Operativos, Tecnicatura Universitaria en Programación a Distancia, UTN Regional San Nicolás.

**Enferrel, A. (s.f.).** *Virtualización de recursos por un hipervisor tipo 2.* Apuntes de clase del curso Arquitectura y Sistemas Operativos, Tecnicatura Universitaria en Programación a Distancia, UTN Regional San Nicolás.

*Docker vs Virtualización.* (s.f.). Apuntes de clase del curso Arquitectura y Sistemas Operativos, Tecnicatura Universitaria en Programación a Distancia, UTN Regional San Nicolás.

*Hipervisor tipo 1 y tipo 2.* (s.f.). Apuntes de clase del curso Arquitectura y Sistemas Operativos, Tecnicatura Universitaria en Programación a Distancia, UTN Regional San Nicolás.

**Amazon Web Services. (s.f.).** *¿Qué es la virtualización?*  
<https://aws.amazon.com/es/what-is/virtualization/>