UNIVERSIDAD TECNOLÓGICA NACIONAL

Trabajo Práctico – Virtualización con VirtualBox

Alumnos:

Juan Ignacio Resende – juani300199@gmail.com

Ignacio Agustin Tello Salas –

Ignacio.tello.dev@gmail.com **Materia:** Arquitectura y

Sistemas Operativos

Profesor: Osvaldo Falabella

Tutora: Ana Valeria Celerier

Fecha de Entrega: 05/06/25

1. Introducción

La virtualización permite ejecutar múltiples sistemas operativos sobre una única máquina física. En este trabajo se exploran conceptos básicos de virtualización y se realiza una instalación y configuración práctica utilizando Oracle VirtualBox. A su vez, se desarrolló un pequeño programa en Python como parte de la práctica final.

2. Marco Teórico

¿Qué es la virtualización?

La virtualización es una tecnología que permite crear representaciones virtuales de recursos físicos, como sistemas operativos, servidores, almacenamiento o redes. En el contexto informático, hace posible ejecutar múltiples máquinas virtuales (VMs) dentro de un mismo equipo físico.

Cada una de estas máquinas virtuales funciona como si fuera una computadora independiente, con su propio sistema operativo, memoria, disco y procesador, aunque

en realidad comparten el hardware de la máquina anfitriona.

Esta capacidad de simular entornos completos dentro de un solo equipo físico ha transformado profundamente el mundo del desarrollo de software, la administración de sistemas y la infraestructura de servidores.

¿Para qué sirve la virtualización?

La virtualización tiene numerosos usos prácticos y beneficios tanto en entornos personales como empresariales. Algunos de sus principales objetivos y ventajas son:

- **Optimización de recursos**: permite utilizar de forma más eficiente el hardware, reduciendo costos.

- **Aislamiento**: cada máquina virtual está separada de las demás, lo que mejora la seguridad.

- **Pruebas y desarrollo**: facilita la creación de entornos de prueba sin afectar el sistema principal.

- **Compatibilidad**: permite correr sistemas operativos antiguos o distintos sin cambiar de hardware.

- **Escalabilidad**: en entornos empresariales, permite implementar múltiples servicios en un mismo servidor físico.

En resumen, la virtualización es clave para el trabajo moderno en tecnología, ya que brinda **flexibilidad, ahorro de recursos y control**.

¿Qué es un hipervisor?

El hipervisor es el componente central que hace posible la virtualización. Es un software que permite crear, gestionar y ejecutar máquinas virtuales, y se encarga de distribuir los recursos físicos del sistema entre las VMs, como la CPU, la memoria RAM, el disco y la red.

Existen dos tipos principales de hipervisores:

- **Tipo 1 (bare-metal)**: se ejecutan directamente sobre el hardware, sin sistema

operativo anfitrión. Son más eficientes y usados principalmente en servidores. Ejemplos: VMware ESXi, Microsoft Hyper-V Server, Xen, KVM.

- **Tipo 2 (hosted)**: funcionan sobre un sistema operativo anfitrión, como Windows o Linux. Son más accesibles para usuarios comunes y desarrolladores. Ejemplos: VirtualBox, VMware Workstation, Parallels.

Cada uno tiene sus ventajas: los de tipo 1 ofrecen mayor rendimiento y estabilidad, mientras que los de tipo 2 son más fáciles de instalar y usar en entornos de prueba o aprendizaje.

Gestión de recursos en máquinas virtuales

Uno de los aspectos clave de la virtualización es cómo se gestionan los recursos del sistema físico. El hipervisor es el encargado de repartir y controlar estos recursos entre las máquinas virtuales activas, para que todas funcionen correctamente sin interferirse.

Los principales recursos que el hipervisor gestiona son:

- El **procesador (CPU)**
- La **memoria RAM**
- El **almacenamiento en disco**
- (Opcionalmente también la red y periféricos)

Cada uno de estos recursos requiere técnicas específicas para ser virtualizado de forma eficiente.

Gestión del procesador (CPU)

El hipervisor se encarga de asignar el uso del procesador físico entre las distintas máquinas virtuales. Para lograrlo, utiliza técnicas como:

- **Segmentación temporal (Time slicing)**: cada VM recibe pequeños turnos de tiempo para acceder al procesador, de forma alternada, lo que permite que varias VMs compartan la misma CPU.

- **CPU Pinning** (en hipervisores avanzados): se asigna un núcleo físico específico del procesador a una máquina virtual, útil en entornos donde se necesita rendimiento constante.

Cada máquina virtual puede tener múltiples núcleos virtuales, pero todos dependen de los núcleos reales del sistema anfitrión.

Gestión de la memoria virtual

La memoria RAM debe gestionarse cuidadosamente para evitar conflictos entre máquinas virtuales. El hipervisor se encarga de simular que cada VM tiene su propia memoria, y lo hace utilizando dos técnicas principales:

- **Shadow Page Tables (SPT)**: el hipervisor mantiene una tabla “sombra” que traduce las direcciones de memoria virtual de la VM a direcciones reales de la memoria física. Este proceso implica intervención constante del hipervisor, lo que puede afectar el rendimiento.

- **Second Level Address Translation (SLAT)**: técnica asistida por hardware que permite a la CPU realizar directamente esta traducción, reduciendo la carga sobre el hipervisor. Mejora notablemente el rendimiento y es utilizada por procesadores modernos con soporte para virtualización (Intel VT-x, AMD-V).

Ambas técnicas permiten que las VMs utilicen la memoria de forma segura, sin interferir con otras.

Gestión del almacenamiento (disco)

Cada máquina virtual utiliza un archivo especial que actúa como si fuera un disco duro real. Este archivo se conoce como **disco virtual**, y puede tener distintos formatos como:

- **VDI** (VirtualBox)

- **VMDK** (VMware)

- **VHD** (Microsoft)

El hipervisor se encarga de gestionar estos discos virtuales, permitiendo configurar:

- Si el disco crece de forma dinámica o tiene tamaño fijo.
- El tipo de conexión simulada (IDE, SATA, SSD).
- El almacenamiento total disponible para cada VM.

Desde dentro de la máquina virtual, el sistema operativo interpreta este archivo como un disco físico completamente funcional.

La virtualización es una herramienta fundamental en la informática moderna. Gracias a ella, es posible ejecutar múltiples entornos aislados dentro de una sola máquina física, optimizando recursos, facilitando el desarrollo y mejorando la seguridad.

El hipervisor es el software clave que hace esto posible, distribuyendo recursos como CPU, memoria y almacenamiento entre las distintas máquinas virtuales.

Comprender cómo se gestionan estos recursos permite usar la virtualización de manera más eficiente y profesional, tanto en laboratorios de prueba como en entornos de producción.

3. Caso Práctico

Para este trabajo se utilizó Oracle VirtualBox, que es un programa que permite crear máquinas virtuales dentro del mismo sistema operativo, sin necesidad de modificar nada en la computadora principal. En este caso, el objetivo era instalar Ubuntu Server 22.04 en una máquina virtual y configurar un servidor web Apache accesible desde la red local.

Primero, se creó una nueva máquina virtual dentro de VirtualBox. Se le puso el nombre “tp´seguridad” para poder identificarla fácilmente. Como sistema operativo se eligió Linux, específicamente Ubuntu (64-bit), y se le asignaron 2067 MB de memoria RAM, lo cual es suficiente para un entorno de servidor básico. También se creó un disco virtual de 20 GB en formato VDI con almacenamiento dinámico, lo que permite que ocupe

espacio solo a medida que se va utilizando.

Luego se descargó la imagen ISO oficial de Ubuntu Server desde el sitio oficial:

🔗 https://ubuntu.com/download/server

Esa imagen se montó en la máquina virtual para iniciar el proceso de instalación. Se eligió la instalación mínima de Ubuntu Server, sin entorno gráfico, y se creó un usuario y contraseña para poder acceder al sistema una vez instalado. En el paso de red, se configuró el uso de DHCP para que la máquina virtual obtenga su dirección IP automáticamente. Además, se activó el modo “Bridge” en la configuración de VirtualBox, lo que permite que la máquina virtual se comporte como si fuera una computadora más dentro de la red local, con acceso a internet y posibilidad de ser accedida desde otros dispositivos conectados a la misma red.

Una vez finalizada la instalación, se inició el sistema y se trabajó desde la terminal. Primero se actualizó la lista de paquetes usando el comando sudo apt update, y luego se instaló el servidor web Apache con sudo apt install apache2. Para permitir el acceso desde otros dispositivos, se configuró el firewall utilizando sudo ufw allow 'Apache', lo que habilita automáticamente el puerto 80 para tráfico HTTP. Para comprobar que funcionaba, se ingresó la dirección IP de la máquina virtual desde un navegador web en otra computadora y se cargó correctamente la página de bienvenida de Apache.

Finalmente, se verificó que Python estuviera instalado utilizando el comando python3 --version, lo cual confirmó que ya venía preinstalado con el sistema. Con eso, quedó listo el entorno para crear y ejecutar scripts. Como parte del trabajo, creé un pequeño programa en Python para probar que todo funcionara correctamente. Usé el editor de texto nano desde la terminal, guardé el archivo con el nombre promedio.py, lo ejecuté desde consola y comprobé que el sistema respondía como se esperaba.

El código que hice fue el siguiente:


```
juan-resende@juan-resende-VirtualBox:~$ nano promedio.py
juan-resende@juan-resende-VirtualBox:~$ python3 promedio.py
ingrese 5 numeros :
ingrese el numero 1 :9
ingrese el numero 2 :9
ingrese el numero 3 :7
ingrese el numero 4 :8
ingrese el numero 5 :9
el promedio es : 8.4
juan-resende@juan-resende-VirtualBox:~$
```

5. Resultados Obtenidos

La máquina virtual con Ubuntu Server se ejecutó correctamente dentro de VirtualBox, sin presentar errores durante el arranque ni durante la instalación de los componentes necesarios. Una vez instalado el sistema operativo, se comprobó que el servidor Apache estuviera funcionando correctamente. Para eso, se ingresó la dirección IP de la máquina virtual en un navegador desde otra computadora conectada a la misma red, y se pudo visualizar sin problemas la página por defecto de Apache, lo que confirmó que el servicio estaba activo.

Además, se verificó que el servicio web fuera accesible desde la red local gracias a la configuración en modo Bridge y al uso de DHCP. También se comprobó que el acceso estuviera protegido correctamente por el firewall configurado en Ubuntu, que solo permitía las conexiones necesarias para el servidor web, brindando así una primera capa de seguridad.

En todo este proceso, se logró comprender de forma práctica el ciclo completo que implica virtualizar un servidor: desde la instalación del sistema operativo, la configuración de servicios como Apache, la habilitación del acceso desde otras máquinas, hasta la protección con firewall. Esta experiencia permitió ver de manera concreta cómo se implementa y gestiona un servidor virtualizado dentro de un entorno seguro y funcional.

6. Conclusiones

Nos pareció que trabajar con virtualización fue una forma muy práctica de aprender a instalar y configurar sistemas sin necesidad de tener varios equipos físicos. Poder crear una máquina virtual y hacer pruebas ahí adentro nos ayudó a entender mejor cómo funciona un servidor, sin correr riesgos en el sistema operativo que usamos todos los días.

VirtualBox nos resultó una herramienta bastante accesible para empezar. Tiene muchas opciones de configuración, pero es fácil de usar. Pudimos instalar Ubuntu Server, configurar Apache y acceder al servidor desde otra máquina por red sin problemas. También nos sirvió para aprender a manejar el firewall y permitir solo lo necesario para que el servidor esté protegido.

Con este trabajo entendimos todo el proceso: desde la instalación de un sistema operativo en una máquina virtual, hasta la configuración de servicios, el acceso desde la red y las medidas básicas de seguridad. Nos pareció muy útil para tener una idea más concreta y práctica de cómo se trabaja realmente cuando se administra un servidor o se configura un entorno de red.

7. Bibliografía

- Video tutorial de la instalacion de ubuntu: https://youtu.be/XfCaHcOdBL8?si=6HTMNq4rFgFgAuL1&utm_source=ZTQx

O • Video explicativo del Profesor Andres Odiard: https://youtu.be/NeIS7m9HEw?si=bIyhUQXSg86JbvDk

• Video explicativo del Profesor Miguel Tola:

https://youtu.be/EjySRGAz9CI?si=NnBWEaG5_J06ZbZO

• Modelo integrador virtualizacion.

• Get Ubuntu Server: Get Ubuntu Server | Download | Ubuntu •

Download VirtualBox: https://www.virtualbox.org/wiki/Downloads

Video explicativo grupal:

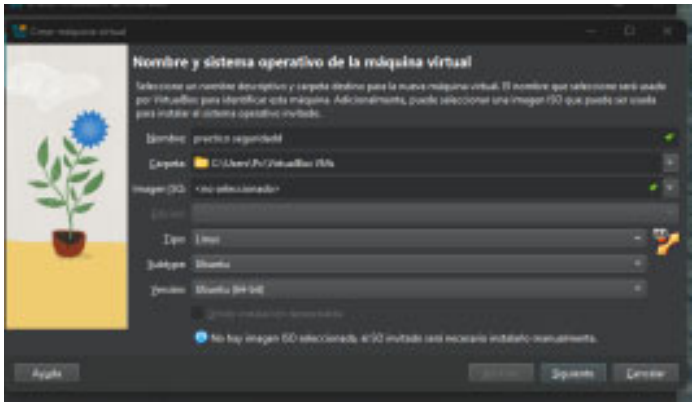
https://youtu.be/BkdX06RkK0M?si=vR8OjY4WcPS1GZ_D

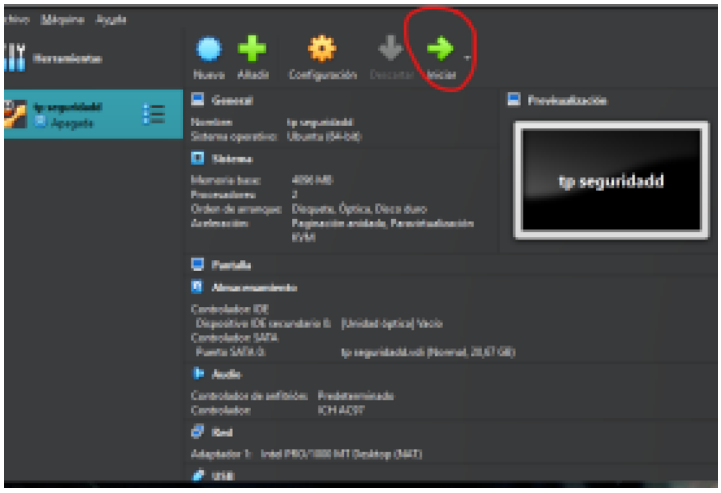
Link de github:

https://github.com/JuanResende/TP-INTEGRADOR-A.Y.S.O-.git

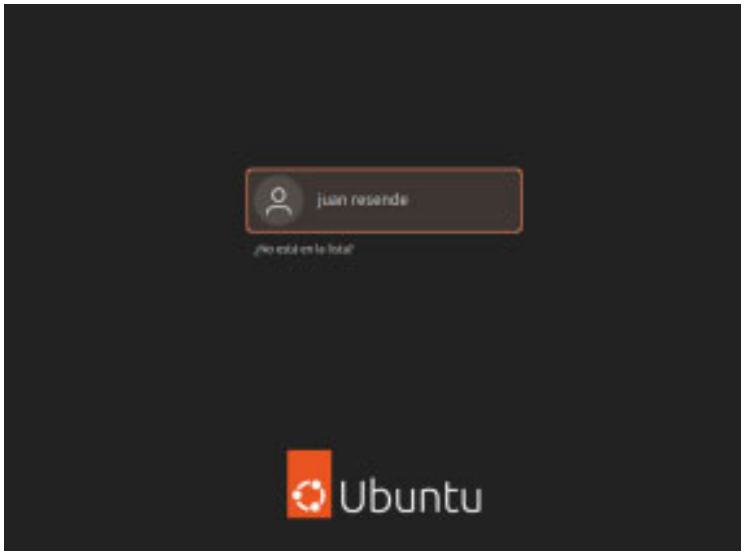
8. Anexos:

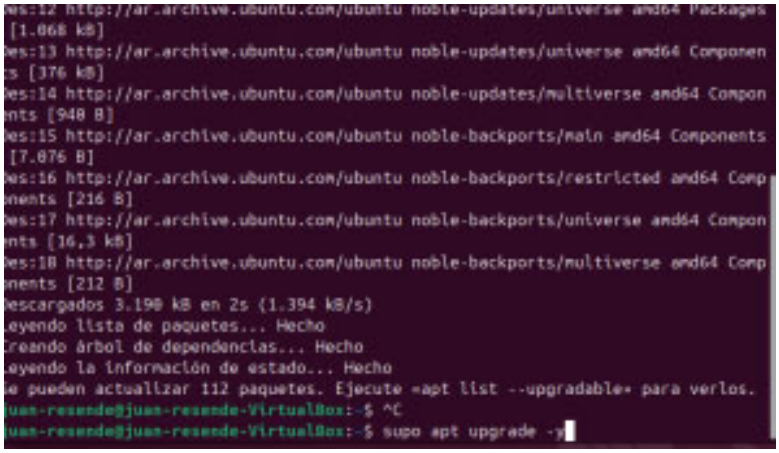
Imagenes del proceso de la creación de la VM:





Una vez iniciado creamos usuario y contraseña:





```
0-1ubuntu7) ...
reparando para desempaquetar .../57-plymouth-theme-spinner_24.004.60-1ubuntu7.1
_amd64.deb ...
esempaquetando plymouth-theme-spinner (24.004.60-1ubuntu7.1) sobre (24.004.60-1
ubuntu7) ...
reparando para desempaquetar .../58-plymouth-label_24.004.60-1ubuntu7.1_amd64.d
eb ...
esempaquetando plymouth-label (24.004.60-1ubuntu7.1) sobre (24.004.60-1ubuntu7)
...
reparando para desempaquetar .../59-plymouth_24.004.60-1ubuntu7.1_amd64.deb ...
esempaquetando plymouth (24.004.60-1ubuntu7.1) sobre (24.004.60-1ubuntu7) ...
reparando para desempaquetar .../60-bluez-cups_5.72-0ubuntu5.1_amd64.deb ...
esempaquetando bluez-cups (5.72-0ubuntu5.1) sobre (5.72-0ubuntu5) ...
reparando para desempaquetar .../61-bluez-obexd_5.72-0ubuntu5.1_amd64.deb ...
esempaquetando bluez-obexd (5.72-0ubuntu5.1) sobre (5.72-0ubuntu5) ...
reparando para desempaquetar .../62-dns-root-data_2024071001-ubuntu0.24.04.1_al
l.deb ...
esempaquetando dns-root-data (2024071801-ubuntu0.24.04.1) sobre (2023112702-wil
sync1) ...

rogreso: [ 32%] [#####################.....................................]
```

```
uan-resende@juan-resende-VirtualBox:~$ ip a
: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
 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 gr
up default qlen 1000
    link/ether 08:00:27:2a:98:c8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.189/24 brd 192.168.100.255 scope global dynamic noprefixrou
te enp0s3
       valid_lft 86355sec preferred_lft 86355sec
    inet6 2803:4fc0:8:c8ea:bd96:ff5b:be5b:afc2/64 scope global temporary dynamic

       valid_lft 259156sec preferred_lft 85866sec
    inet6 2803:4fc0:8:c8ea:a08:27ff:fe2a:98c8/64 scope global dynamic mngtmpaddr

       valid_lft 259156sec preferred_lft 172756sec
    inet6 fe80::a08:27ff:fe2a:98c8/64 scope link
       valid_lft forever preferred_lft forever
uan-resende@juan-resende-VirtualBox:~$
```