

Trabajo Práctico – Virtualización con VirtualBox

Alumnos:

Matias Giraldo

Marcos Glocker

Materia: Arquitectura y Sistemas Operativos

Fecha de entrega: 05 de junio de 2025

Índice

1. Introducción
2. Marco Teórico
3. Caso Práctico
4. Metodología Utilizada
5. Resultados Obtenidos
6. Conclusiones
7. Bibliografía

Introducción

La virtualización es una tecnología que permite ejecutar múltiples sistemas operativos en una sola máquina física, optimizando el uso de recursos y facilitando tareas como desarrollo, pruebas o simulaciones. En este trabajo práctico se exploran los conceptos fundamentales de la virtualización y se realiza una instalación y configuración básica de una máquina virtual utilizando Oracle VirtualBox. Además, se ejecuta un script en Python como ejemplo de uso práctico del entorno virtualizado.

Marco Teórico

¿Qué es la virtualización? ¿Cuáles son sus beneficios?

- La virtualización es una tecnología que se puede usar para crear representaciones virtuales de recursos informáticos, como servidores, almacenamiento, redes u otras máquinas físicas. Sus beneficios son la eficiencia, flexibilidad, ahorro de costos y escalabilidad.

¿Qué es una máquina virtual (VM)?

- Una máquina virtual es un equipo definido por software que se ejecuta en un equipo físico con un sistema operativo y recursos informáticos independientes. Una computadora física se denomina host, en cambio, las máquinas virtuales son máquinas invitadas.

¿Qué es VirtualBox?

- VirtualBox es un software de virtualización multiplataforma de código abierto, permite a los desarrolladores entregar código más rápido, ya que pueden ejecutar múltiples sistemas operativos en un solo dispositivo.

Tipos de virtualización:

- Virtualización de servidores
- Virtualización de almacenamiento
- Virtualización de red
- Virtualización de datos
- Virtualización de aplicaciones
- Virtualización de escritorios

¿Qué es un hipervisor? ¿Qué tipos existen? ¿Cuál es la diferencia entre los tipos?

- Un hipervisor es un componente de software que permite crear y gestionar máquinas virtuales. Actúa como intermediario entre el hardware físico y los SO que se ejecuta en las VMs (asegura que cada VM tenga acceso controlado a los recursos evitando que una VM interfiera con el funcionamiento de otras VMs).

- Existen el hipervisor de tipo 1 y de tipo 2.

Aspecto	Hypervisor Tipo 1 (Bare-metal)	Hypervisor Tipo 2 (Alojado)
Control sobre el hardware	Directo: el hypervisor tiene control completo sobre la CPU, memoria y dispositivos.	Indirecto: depende del sistema anfitrión para acceder al hardware.
Rendimiento	Alto, similar al hardware nativo.	Más bajo, debido a la capa adicional del sistema anfitrión.
Seguridad	Mayor: las VMs están aisladas directamente del hardware.	Menor: vulnerabilidades en el sistema anfitrión pueden afectar a las VMs.
Eficiencia	Utiliza los recursos del hardware de manera óptima.	Menos eficiente, ya que el sistema anfitrión también consume recursos.

¿Qué es Docker?

- Docker es una tecnología de organización en contenedores que posibilita la creación y el uso de los contenedores de Linux. Se pueden utilizar contenedores como VMs muy livianas y modulares, y obtiene la flexibilidad necesaria para crearlos, implementarlos, copiarlos y trasladarlos de un entorno a otro.

¿Qué es un contenedor?

- Un contenedor es un software que agrupa el código de una aplicación, sus dependencias y configuraciones necesarias para que funcione en cualquier entorno.

¿Qué es una imagen?

- Una imagen es una plantilla de solo lectura que contiene el sistema de archivos, el código y las dependencias necesarias para ejecutar un contenedor. Los contenedores se crean a partir de estas.

¿Qué es un volumen?

- Un volumen es un sistema de almacenamiento, que permite guardar datos fuera del ciclo de vida del contenedor. De esta forma, los datos no se pierden cuando el contenedor se elimina o reinicia.

¿Para qué se usa docker compose?

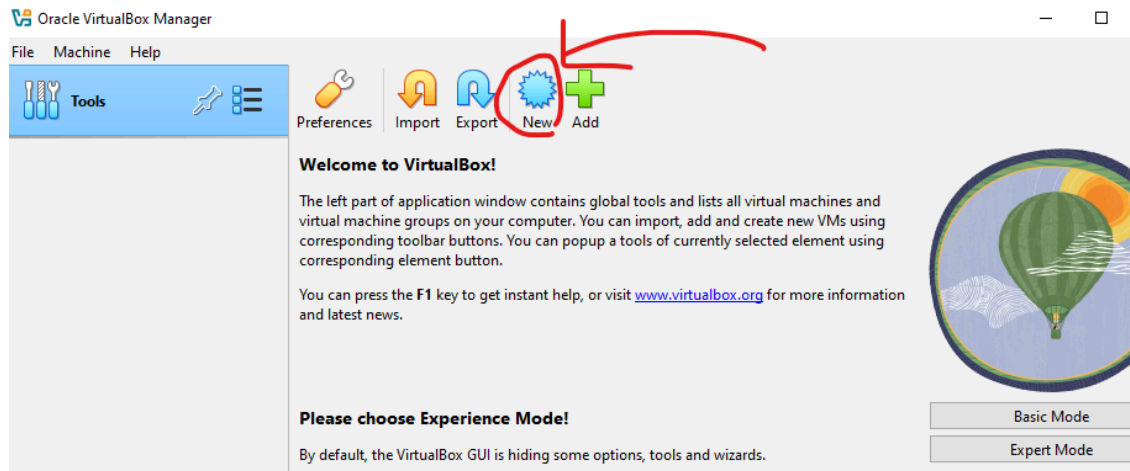
- Docker Compose es una herramienta versátil que permite definir y gestionar aplicaciones multi-contenedor de forma sencilla. Se puede describir la configuración del entorno de desarrollo en un archivo YML, especificando los servicios, volúmenes y redes necesarios para la aplicación. Luego, con un solo comando, puedes crear y ejecutar todos los contenedores definidos en tu archivo de configuración.

Caso Práctico

En un entorno, es posible tener la necesidad de ejecutar programas/scripts sin afectar al SO principal. Lo que podría significar riesgos de seguridad, estabilidad, etc. para el equipo anfitrión. Para resolver el problema, recurrimos a la virtualización, que nos permite crear un entorno aislado donde podremos instalar un SO independiente y ejecutar el código Python de manera segura y controlada.

Habiendo descargado e instalado VirtualBox e imagen ISO de Ubuntu pasamos a la creación de una máquina virtual en VirtualBox.

Creación de una nueva VM en VirtualBox:



Le asignamos nombre, la dirección donde se guardará y la imagen ISO (Ubuntu).

Virtual machine Name and Operating System


Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: ✓

Folder: ▾

ISO Image: ✓ ▾


Edition: ▾

Type: ▾ 

Subtype: ▾

Version: ▾

☐ Skip Unattended Installation

 Detected OS type: Ubuntu (64-bit). This OS type can be installed unattendedly. The install will start after this wizard is closed.

Asignamos un nombre de usuario y contraseña.

Unattended Guest OS Install Setup

You can configure the unattended guest OS install by modifying username, password, and hostname. Additionally you can enable guest additions install. For Microsoft Windows guests it is possible to provide a product key.

Username and Password	Additional Options
Username: TP-integrador ✓	Product Key: #####-#####-#####-#####-#####
Password: ●●●●●● ✓	Hostname: SimulacionTP ✓
Repeat Password: ●●●●●● ✓	Domain Name: myguest.virtualbox.org ✓
	<input type="checkbox"/> Install in Background

☐ Guest Additions

Guest Additions ISO: C:\Program Files\Oracle\VirtualBox\VBGuestAdditions.iso

Back Next Cancel

Asignamos los recursos de hardware que ocupará.

Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory: 2048 MB

Processors: 2

☐ Enable EFI (special OSes only)

Back Next Cancel

Y asignamos el almacenamiento que dispondrá la VM.

Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

☒ Create a Virtual Hard Disk Now

Disk Size: 20.00 GB

4.00 MB 2.00 TB

☐ Pre-allocate Full Size

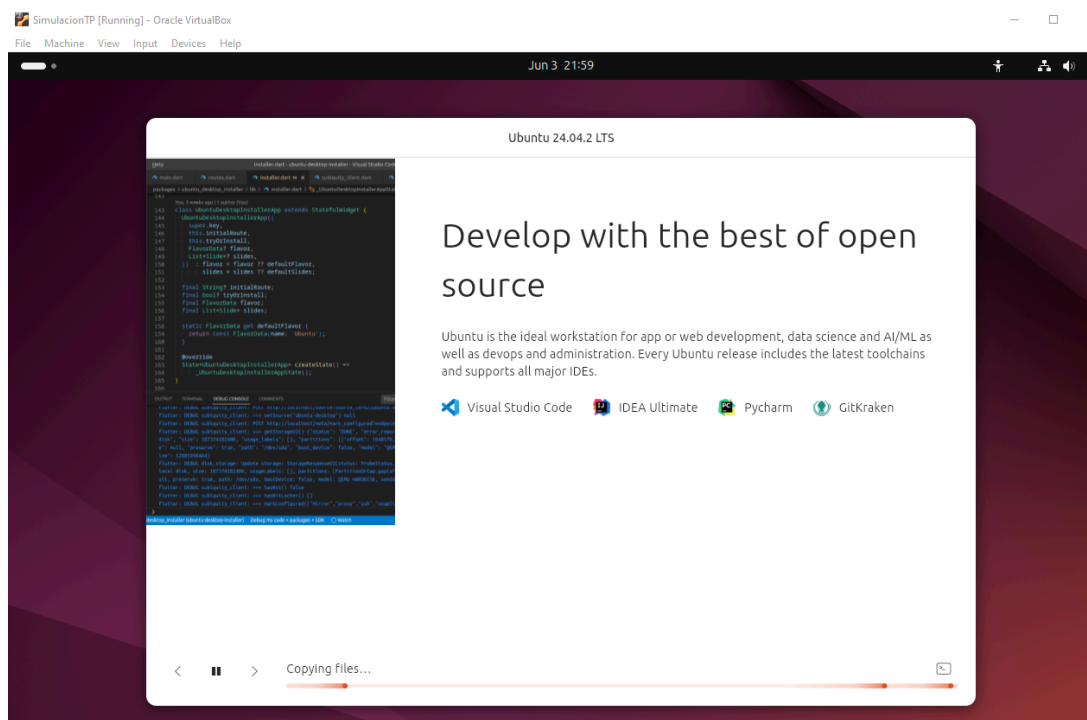
☐ Use an Existing Virtual Hard Disk File

Empty

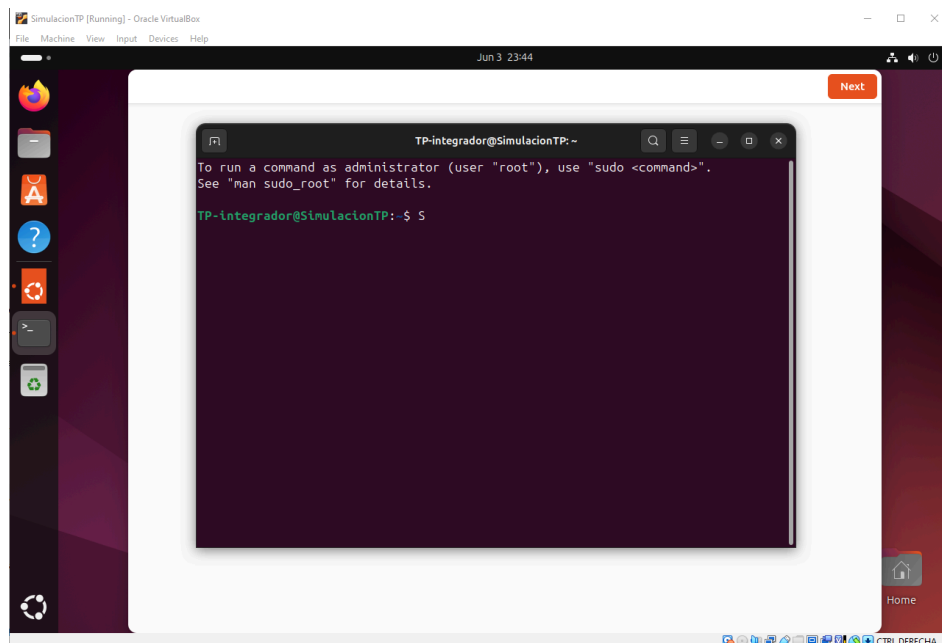
☐ Do Not Add a Virtual Hard Disk

Back Next Cancel

Arrancamos la VM (Se copian archivos)



Ingresamos a la consola (Ctrl + Alt + T)



Verificamos si tenemos Python instalado primero.

```
TP-integrador@SimulacionTP: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

TP-integrador@SimulacionTP:~$ python3 --version
Python 3.12.3
TP-integrador@SimulacionTP:~$
```

Creamos el script:

```
TP-integrador@SimulacionTP:~$ nano programa.py

print("Ingrese 3 numeros para obtener el promedio.")

lista=["","",""]

for i in range(0,3):
    lista[i]=int(input(f"Ingrese numero {i+1}: "))

prom= (lista[0] + lista[1] + lista[2])/3

print(f"EL promedio de los 3 numeros es {round(prom,2)}")
```

Lo ejecutamos e interactuamos con el:

```
TP-integrador@SimulacionTP:~$ python3 programa.py
Ingrese 3 numeros para obtener el promedio.
Ingrese numero 1: 4
Ingrese numero 2: 6
Ingrese numero 3: 8
EL promedio de los 3 numeros es 6.0
TP-integrador@SimulacionTP:~$
```

Metodología Utilizada

Los pasos que se realizaron fueron los siguientes:

- Descargamos e instalamos VirtualBox y Ubuntu.
- Creamos una nueva VM
- Configuramos la VM (nombre, imagen, asignación de recursos, etc.)
- Iniciamos la VM y esperamos que inicie
- Una vez iniciada, ingresamos a la consola de la VM.
- Creamos el script de Python y le copiamos el código fuente desarrollado en VS.
- Finalmente, ejecutamos e interactuamos con el programa.

La investigación previa hecha fue leer toda la documentación que se encontraba en el aula. Después tuvimos algunas dudas o queríamos profundizar más y buscamos para complementar la información.

Utilizamos Visual Studio para diseñar, desarrollar y probar el código para luego pasarlo a un archivo ejecutable en la VM. Utilizamos GitHub para guardar el proyecto en la nube.

Uno se encargó más de la presentación del TP para que cumpla con la estructura recomendada y el otro se encargó más de buscar información de distintas fuentes en internet que podrían servir para afianzar mejor los conceptos de Virtualización.

Se realizó un código sencillo por lo que no necesito de etapas de prueba como tal, obviamente, hicimos implementaciones de las herramientas o estructuras que se pedían (listas, bucles, etc.)

Resultados obtenidos

Logramos instalar y configurar de manera exitosa una máquina virtual con Ubuntu en VirtualBox. Simulamos un entorno de trabajo independiente de nuestro sistema operativo principal, lo cual es lo ideal para hacer pruebas para que no comprometa al equipo anfitrión.

La instalación del software (VirtualBox, Ubuntu) y la creación de la VM se realizaron sin problemas. Pudimos configurar y ejecutar correctamente la imagen ISO de Linux.

Accedimos a la consola de la VM sin inconvenientes. Y luego, creamos y ejecutamos el script de Python.

Tuvimos una dificultad en la carga de la VM, debido a la asignación de recursos (le habíamos asignado 2048 MB de memoria RAM), pero pudimos solucionarlo agregando más memoria RAM (le pusimos 4096 MB).

Verificamos que Python ya estuviera instalado con "python3 --version" para, si en caso de tenerlo, no volver a instalarlo.

Conclusiones

Se comprendieron los conceptos fundamentales de la virtualización y se experimentó con su aplicación mediante VirtualBox. Se aprendió a crear, configurar y operar una máquina virtual con Ubuntu, así como también a ejecutar scripts de Python en un entorno aislado. Esto nos permitió darnos cuenta de la importancia de las VMs para pruebas de software, entornos seguros y la administración de los recursos.

Una posible mejora quizás hubiese sido haber creado el entorno también en docker y comparar el rendimiento entre VM y contenedores.

Bibliografía

Información adicional sobre la virtualización:

<https://aws.amazon.com/es/what-is/virtualization/>

<https://www.oracle.com/ar/virtualization/virtualbox/>

Información adicional sobre Docker y Docker Compose:

<https://www.redhat.com/es/topics/containers/what-is-docker>

<https://imaginaformacion.com/tutoriales/que-es-docker-compose>

Documentación de VirtualBox:

<https://www.virtualbox.org/wiki/Documentation>