

Trabajo Práctico Integrador

Virtualización de un entorno Linux y desarrollo de un programa en Python usando VirtualBox

Alumnos:

- Luciano José Cartagena– lucianocartagena17@gmail.com
- Santiago Arroquigaray– arroqui192@gmail.com

Materia: Arquitectura y Sistemas Operativos

Profesor/a: Martin Aristiaran

Fecha de Entrega: 05/06/2025

Índice

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

1. Introducción

En este trabajo integrador se explora la virtualización como herramienta clave en la administración de sistemas y el desarrollo de software. Se utiliza VirtualBox, un software

gratuito y multiplataforma, para crear y gestionar una máquina virtual Linux dentro de un sistema operativo anfitrión.

El objetivo principal es instalar y configurar una distribución Linux, desarrollar y ejecutar un programa sencillo en Python dentro de la máquina virtual, y documentar todo el proceso. Este enfoque permite reforzar competencias técnicas y metodológicas fundamentales para la carrera, proporcionando experiencia práctica en entornos virtualizados que son ampliamente utilizados en la industria tecnológica.

La virtualización facilita la experimentación, el desarrollo y las pruebas en ambientes seguros y controlados, sin afectar el sistema operativo principal del equipo. Esta habilidad es muy valorada en el ámbito profesional, ya que permite crear entornos aislados para pruebas, desarrollo y despliegue de aplicaciones.

2. Marco Teórico

La virtualización es una tecnología que permite simular hardware y sistemas completos dentro de un sistema operativo, creando máquinas virtuales (VMs) que funcionan como computadoras independientes. Esta simulación se realiza dentro de un entorno aislado o sandbox, donde los cambios y pruebas no afectan al sistema principal.

Ventajas de la Virtualización:

- Permite experimentar y probar software sin riesgos para el sistema principal
- Facilita la administración de recursos y la creación de entornos de prueba
- Es fundamental en la industria para desarrollo, testing y despliegue seguro
- Optimiza el uso de recursos de hardware
- Permite ejecutar múltiples sistemas operativos simultáneamente

VirtualBox:

VirtualBox es un hipervisor de tipo 2, desarrollado por Oracle, gratuito y multiplataforma. Actúa como un administrador de máquinas virtuales, asignando recursos como CPU, memoria RAM, disco y gestionando su interacción con el hardware físico. Su facilidad de uso y compatibilidad con diversos sistemas operativos lo convierten en una herramienta ideal para el aprendizaje y la experimentación.

Conceptos Clave:

- Sandbox: Un entorno aislado donde los procesos no afectan al sistema anfitrión, ideal para pruebas y aprendizaje seguro
- Hipervisor: Software que permite crear y gestionar máquinas virtuales
- Imagen ISO: Archivo que contiene una copia completa de un sistema de archivos, utilizado para instalar sistemas operativos

- Recursos virtuales: CPU, RAM, disco y red asignados a cada máquina virtual

3. Objetivos

Este trabajo integrador tiene como propósito desarrollar competencias prácticas en virtualización y programación, estableciendo los siguientes objetivos específicos:

1. Instalar y configurar VirtualBox en el sistema operativo anfitrión
2. Crear una máquina virtual con una distribución Linux (Linux Mint o Ubuntu Server)
3. Instalar y configurar Python en la máquina virtual
4. Desarrollar y ejecutar un programa sencillo en Python dentro de la VM
5. Documentar el proceso completo con capturas de pantalla y evidencias
6. Crear un repositorio online con todo el material del proyecto
7. Generar un video explicativo demostrando el funcionamiento del sistema

4. Caso Práctico

El caso práctico consistió en la implementación completa de un entorno virtualizado para desarrollo en Python, siguiendo una metodología estructurada y documentando cada etapa del proceso.

Pasos Realizados:

4.1 Instalación de VirtualBox

Se descargó e instaló VirtualBox desde el sitio oficial ([virtualbox.org](https://www.virtualbox.org)), eligiendo la versión adecuada para el sistema operativo anfitrión. Durante la instalación se configuraron las interfaces de red virtuales necesarias.

4.2 Creación de la Máquina Virtual

Se configuró una nueva máquina virtual con las siguientes especificaciones:

- Nombre descriptivo: "Linux-Python-VM"
- Sistema operativo: Linux Mint 21.3
- Memoria RAM: 2 GB
- Disco virtual: 20 GB (dinámicamente asignado)

- Configuración de red: NAT para acceso a internet

4.3 Instalación del Sistema Operativo

Se descargó la imagen ISO de Linux Mint desde el sitio oficial, se montó en VirtualBox y se siguió el asistente de instalación. Se configuró el usuario administrador y se completó la instalación básica del sistema.

4.4 Configuración de Python

Se verificó la versión de Python instalada y se actualizó el sistema:

```
python3 --version
sudo apt update
sudo apt upgrade
sudo apt install python3-pip
```

4.5 Desarrollo del Programa en Python

Se creó un programa que solicita tres notas al usuario, calcula el promedio y muestra el resultado. El código implementado fue:

```
# Programa para calcular promedio de notas
print("=== Calculadora de Promedio de Notas ===")
notas = []

for i in range(3):
    while True:
        try:
            nota = float(input(f"Ingrese la nota {i+1} (0-10): "))
            if 0 <= nota <= 10:
                notas.append(nota)
                break
            else:
                print("La nota debe estar entre 0 y 10")
        except ValueError:
            print("Por favor ingrese un número válido")

promedio = sum(notas) / len(notas)
print(f"\nNotas ingresadas: {notas}")
print(f"El promedio es: {promedio:.2f}")

if promedio >= 7:
    print("¡Excelente trabajo!")
elif promedio >= 5:
    print("Buen trabajo, pero se puede mejorar")
else:
    print("Es necesario estudiar más")
```

4.6 Pruebas y Validación

Se ejecutó el programa múltiples veces con diferentes valores de entrada para verificar su correcto funcionamiento. Se tomaron capturas de pantalla de la ejecución y los resultados obtenidos.

5. Metodología Utilizada

- Investigación previa sobre virtualización y uso de VirtualBox mediante documentación oficial y recursos académicos
- Instalación y configuración paso a paso, documentando cada etapa con capturas de pantalla
- Desarrollo colaborativo con distribución de tareas: un integrante se encargó de la instalación y configuración técnica, el otro de la documentación y creación del video explicativo
- Uso de control de versiones y repositorio online (GitHub) para organizar y compartir el trabajo
- Aplicación de buenas prácticas de programación: código comentado, manejo de errores y validación de entrada
- Pruebas exhaustivas del programa desarrollado con diferentes casos de uso
- Documentación técnica siguiendo estándares académicos y profesionales

6. Resultados Obtenidos

- Se instaló y configuró correctamente VirtualBox sin problemas de compatibilidad
- La máquina virtual Linux Mint funcionó de manera estable con los recursos asignados
- El programa en Python se ejecutó correctamente dentro del entorno virtualizado
- Se logró una comprensión profunda de los conceptos de virtualización y entornos aislados
- Se documentó exitosamente todo el proceso con capturas de pantalla y evidencias
- Se creó un repositorio organizado con todo el material del proyecto
- Se desarrollaron habilidades de resolución de problemas técnicos de forma autónoma
- Se comprendió la importancia de los entornos virtualizados para el desarrollo seguro

El proceso permitió experimentar con un entorno Linux sin riesgos para el sistema principal, demostrando las ventajas prácticas de la virtualización en el desarrollo de software.

7. Conclusiones

Este trabajo integrador permitió aplicar conocimientos teóricos de virtualización y programación en un entorno práctico real, enfrentando y resolviendo diversas dificultades técnicas que surgieron durante el proceso.

Aprendizajes Obtenidos:

- Dominio de herramientas de virtualización y comprensión de su importancia en la industria
- Experiencia práctica con sistemas operativos Linux y su instalación
- Desarrollo de programas Python en entornos virtualizados
- Habilidades de documentación técnica y organización de proyectos
- Capacidad de resolución autónoma de problemas técnicos

Dificultades Encontradas:

- Adaptación inicial al entorno Linux para usuarios acostumbrados a Windows
- Configuración de recursos de la máquina virtual para optimizar rendimiento
- Manejo de la línea de comandos en Linux
- Organización y estructura del repositorio de código

Propuestas de Mejora:

- Explorar otras distribuciones Linux más especializadas (Ubuntu Server, CentOS)
- Implementar programas Python más complejos con bases de datos o interfaces gráficas
- Utilizar otras herramientas de virtualización como VMware o Docker
- Automatizar la configuración mediante scripts
- Incluir pruebas unitarias en el código desarrollado

La virtualización demostró ser una herramienta fundamental para el desarrollo seguro y la experimentación, proporcionando un entorno controlado que facilita el aprendizaje y la innovación sin comprometer la estabilidad del sistema principal.

8. Bibliografía

- Oracle VirtualBox Manual. (2024). Oracle Corporation. <https://www.virtualbox.org/manual/>
- Linux Mint Documentation. (2024). Linux Mint Team. <https://linuxmint.com/documentation.php>
- Python Official Documentation. (2024). Python Software Foundation. <https://docs.python.org/3/>
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). John Wiley & Sons.
- Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.
- Ubuntu Server Guide. (2024). Canonical Ltd. <https://ubuntu.com/server/docs>

9. Anexos

Los siguientes recursos complementarios están disponibles en el repositorio del proyecto:

- Capturas de pantalla detalladas de cada etapa del proceso de instalación y configuración
- Código fuente completo del programa Python desarrollado (notas.py)
- Archivo README.md con instrucciones detalladas para replicar el proyecto
- Video explicativo de 10-15 minutos demostrando el funcionamiento completo del sistema
- Scripts adicionales para automatización de tareas básicas
- Diagrama de arquitectura del entorno virtualizado implementado

Enlace al repositorio del proyecto: [Insertar URL del repositorio GitHub/GitLab]

Enlace al video explicativo: [Insertar URL del video en YouTube/Vimeo]