



TRABAJO INTEGRADOR – VIRTUALIZACIÓN DE RECURSOS CON HYPERVISOR TIPO 2

Alumnos: [Emanuel Aaron Brahim Pollini (emanuelaaronb@gmail.com)]
[Nadia Celeste Almada (almada.nadia.c@gmail.com)]

Materia: Arquitectura y Sistemas Operativos

Profesor/a: [Diego Lobos y Carlos Andrés Odiard]

Fecha de Entrega: [05/06/2025]

Índice

1. Introducción.....	2
2. Marco Teórico	2
□ Hipervisor Tipo 1 (bare-metal):	2
□ Hipervisor Tipo 2 (hosted):.....	2
3. Caso Práctico	3
Código utilizado:	4
Validación del funcionamiento	4
Capturas de pantalla.....	5
4. Metodología Utilizada	5
5. Resultados Obtenidos	6
Casos de prueba	6
Validación del funcionamiento	7
6. Conclusiones.....	8
7. Bibliografía	8
8. Anexos.....	9

1. Introducción

La virtualización es una tecnología clave en el ámbito de la informática moderna, que permite ejecutar múltiples sistemas operativos en un mismo equipo físico mediante entornos simulados. Esta técnica facilita el desarrollo, la prueba de software y la administración de sistemas sin comprometer el entorno principal del equipo.

El estudio de la virtualización resulta relevante debido a su creciente aplicación en la programación y en la infraestructura tecnológica. Su comprensión es fundamental para la formación del técnico en programación, ya que permite asimilar conceptos esenciales como el uso eficiente de recursos, la compatibilidad entre sistemas y el aislamiento de entornos.

El trabajo tiene como objetivo principal adquirir experiencia en la creación y gestión de máquinas virtuales utilizando VirtualBox, un hipervisor de tipo 2. Además, se propone observar las diferencias de rendimiento al ejecutar un mismo programa tanto en el sistema anfitrión (host) como en el sistema virtualizado (guest), y analizar el impacto de la asignación de recursos, como memoria RAM, núcleos de procesador y espacio en disco, en el funcionamiento general del entorno virtual.

2. Marco Teórico

La virtualización es una técnica que permite ejecutar múltiples sistemas operativos simultáneamente sobre un único equipo físico, creando entornos simulados llamados máquinas virtuales (VM). Cada VM opera de manera aislada, como si fuera un equipo independiente, compartiendo recursos con otras máquinas virtuales y con el sistema operativo anfitrión.

El componente principal que posibilita la virtualización es el **hipervisor**. Según su tipo, los hipervisores se clasifican en:

- **Hipervisor Tipo 1 (bare-metal):**
Se instala directamente sobre el hardware físico, sin un sistema operativo anfitrión, lo que permite una virtualización eficiente y con menor sobrecarga.
- **Hipervisor Tipo 2 (hosted):**
Se ejecuta como una aplicación sobre un sistema operativo anfitrión, utilizando las APIs y servicios del mismo para gestionar recursos y proporcionar un entorno virtualizado a las máquinas virtuales. Es más sencillo de implementar y adecuado para entornos domésticos, educativos o de prueba.

En este trabajo se utilizó VirtualBox, un hipervisor tipo 2, que permite ejecutar sistemas operativos invitados sobre un sistema operativo anfitrión.

La máquina virtual creada ejecuta Ubuntu Desktop, una distribución de GNU/Linux que ofrece entorno gráfico y modo terminal, con bajo consumo de recursos y facilidad de uso, ideal para prácticas en entornos virtualizados.

Además, para el análisis práctico se utilizó Python, un lenguaje de programación interpretado y multiplataforma, para desarrollar un script que muestra el uso de memoria RAM en el sistema anfitrión y en la máquina virtual, permitiendo observar diferencias en la asignación de recursos y su impacto.

Las definiciones y conceptos presentados están basados en la documentación oficial y material teórico provisto en la cursada.

3. Caso Práctico

En este trabajo práctico descargamos e instalamos VirtualBox en un sistema Windows 10, creamos máquinas virtuales configurando los recursos necesarios (RAM, núcleos de procesador y almacenamiento), y luego, dentro del sistema operativo invitado Ubuntu o Lubuntu, instalamos Python desde la terminal para crear y ejecutar un programa que permite comparar la memoria RAM total del equipo, la asignada a la máquina virtual y la memoria disponible restante. Todo lo hicimos exclusivamente desde la terminal del sistema operativo invitado.

Integrante	Actividad realizada	Detalles
Emanuel	Instalación y configuración de VM con Ubuntu Desktop	Descargó VirtualBox desde virtualbox.org y la ISO de Ubuntu 20.04.2-desktop desde ubuntu.com . Configuró la máquina virtual con 6 GB de RAM, 2 núcleos y 25 GB de almacenamiento.
Nadia	Instalación y configuración de VM con Lubuntu Desktop	Descargó la ISO de Lubuntu 24.04.2-desktop desde lubuntu.me . Configuró su VM con 4 GB de RAM, 2 núcleos y 25 GB de almacenamiento.

Ambos instalamos Python3 desde la terminal del sistema operativo invitado y desarrollamos un script en Python llamado ram.py, que calcula el uso de memoria RAM disponible comparando los recursos del equipo con los asignados a la máquina virtual.

Código utilizado:

```
# Se crea una lista para almacenar los valores de RAM

ram_dispositivo = []

# Solicitar la memoria RAM total del equipo

ram_total = int(input("¿Cuánta memoria RAM tiene tu PC en GB? "))

ram_dispositivo.append(ram_total)

# Solicitar la memoria RAM asignada a la máquina virtual

ram_vm = int(input("¿Cuánta memoria RAM le asignaste a la VM en GB? "))

ram_dispositivo.append(ram_vm)

# Calcular la RAM disponible restante

ram_disponible = ram_dispositivo[0] - ram_dispositivo[1]

ram_dispositivo.append(ram_disponible)

# Mostrar resumen del uso de memoria RAM

print("\nResumen del uso de memoria RAM:")

print(f"- RAM total de la PC: {ram_dispositivo[0]} GB")

print(f"- RAM asignada a la máquina virtual: {ram_dispositivo[1]} GB")

print(f"- RAM disponible restante: {ram_dispositivo[2]} GB")
```

Validación del funcionamiento

Probamos el script en ambas máquinas virtuales y funcionó correctamente. El programa pidió los datos, los almacenó y calculó de forma precisa la RAM restante, permitiéndonos comparar el uso de memoria entre el equipo anfitrión y la máquina virtual. Esta práctica nos ayudó a entender mejor cómo afecta la asignación de recursos al rendimiento de los entornos virtualizados.

Capturas de pantalla

Las siguientes capturas se incluyen en la carpeta imagenes/ del proyecto y también están disponibles en el repositorio:

- *Configuración de la máquina virtual en VirtualBox (RAM, núcleos, almacenamiento).*
- *Ejecución del script ram.py en la terminal con su resultado.*

Las imágenes pueden visualizarse en el siguiente enlace:

<https://github.com/EmanuelABP/UTN-TP-Final-Virtualizacion>

4. Metodología Utilizada

Para comenzar, revisamos todo el material teórico brindado en la cursada sobre virtualización, incluyendo los apuntes disponibles en la plataforma y el video explicativo proporcionado por el docente: <https://www.youtube.com/watch?v=Ne-IS7m9HEw>. Esto nos permitió comprender los conceptos básicos y la consigna del trabajo.

Luego avanzamos con la parte práctica, que consistió en los siguientes pasos:

1. Descargamos e instalamos VirtualBox en una computadora con sistema operativo Windows 10.
2. Descargamos una imagen ISO: Ubuntu 20.04.2-desktop en un caso, y Lubuntu 24.04.2-desktop en el otro.
3. Creamos una máquina virtual en cada equipo, asignándole memoria RAM, cantidad de núcleos y espacio en disco.
4. Instalamos el sistema operativo dentro de la máquina virtual.
5. Accedimos a la terminal del sistema operativo invitado.
6. Instalamos Python3 desde la terminal.
7. Creamos un archivo .py con un script que calcula y muestra el uso de memoria RAM.
8. Ejecutamos el programa directamente desde la terminal para verificar su correcto funcionamiento.

Las herramientas y recursos que utilizamos fueron:

- **VirtualBox** como entorno de virtualización.
- **Ubuntu Desktop** y **Lubuntu Desktop** como sistemas operativos invitados.
- **Python 3** como lenguaje de programación.
- **Terminal de Linux** como entorno de trabajo.
- **Material de la materia** como fuente de estudio y referencia.

En cuanto al trabajo en equipo, ambos integrantes colaboramos durante toda la elaboración. Primero investigamos juntos el contenido teórico, compartimos lo que entendimos y resolvimos dudas entre nosotros. Luego, realizamos la práctica por separado en nuestras respectivas computadoras, para asegurarnos de que los dos comprendiéramos todo el proceso y pudiéramos verificar el funcionamiento del script en distintas configuraciones de máquinas virtuales.

5. Resultados Obtenidos

Como resultado del trabajo práctico, instalamos correctamente VirtualBox, creamos nuestras propias máquinas virtuales, instalamos un sistema operativo (Ubuntu o Lubuntu) y ejecutamos un script en Python desarrollado desde la terminal del sistema operativo invitado. Cumplimos con el objetivo principal del trabajo: comprobar que se puede utilizar Python dentro de un entorno virtualizado. El script que usamos fue un ejemplo simple, pensado con fines prácticos, que incluye interacción con el usuario y uso de listas.

Casos de prueba

Cada uno realizó el procedimiento completo en su propia computadora, lo que nos permitió validar de forma individual tanto la instalación como la ejecución del sistema virtualizado. Además, confirmamos el correcto funcionamiento del entorno de Python en ambos sistemas operativos invitados.

Integrante	Resultado General	Errores Detectados y Correcciones
Emanuel	Instalé VirtualBox en Windows 10, descargué Ubuntu 20.04.2 y configuré una VM con 6 GB de RAM, 2 núcleos y 25 GB de disco. Instalé Python3 y ejecuté el script correctamente.	- Intenté instalar VirtualBox en otro disco, pero el instalador arrojó error, así que usé la ruta predeterminada (no tengo captura).- Durante la instalación de Ubuntu, puse "emaue-PC" como nombre de usuario, pero el sistema no permite mayúsculas, así que lo corregí usando minúsculas (hay captura).- Tuve un error de sintaxis en el script Python que logré corregir y ejecutar correctamente (captura en el repositorio).
Nadia	Instalé VirtualBox, descargué Ubuntu 24.04.2 y configuré mi VM con 4 GB de RAM, 2 núcleos y 25 GB de disco. Instalé Python3 y ejecuté el mismo script correctamente.	-Los errores se encuentran en el repositorio con las capturas, comentarios y solución

Validación del funcionamiento

Pudimos crear correctamente nuestras máquinas virtuales con Ubuntu y Ubuntu, instalar Python desde la terminal y ejecutar sin problemas el script. Confirmamos que Python funciona adecuadamente dentro del entorno virtual, y que se pueden desarrollar y ejecutar scripts simples sin inconvenientes. Aunque el script fue un ejemplo básico relacionado con la memoria RAM, su único objetivo fue comprobar el uso de Python, practicar la interacción con el usuario y el manejo de listas.

6. Conclusiones

Con este trabajo aprendimos a descargar e instalar VirtualBox y las imágenes ISO de Ubuntu y Lubuntu. Comprendimos cómo crear una máquina virtual desde cero, asignarle los recursos adecuados y realizar correctamente la instalación de los sistemas operativos. También confirmamos que es posible trabajar únicamente desde la terminal de estos sistemas para instalar Python3, crear un programa y ejecutarlo de manera efectiva.

A lo largo del proceso reforzamos conceptos sobre virtualización, sistemas operativos y el uso de herramientas básicas en entornos Linux. Además, nos dimos cuenta de que no se necesita un entorno gráfico complejo para realizar tareas de programación simples, lo cual es útil en sistemas con pocos recursos.

Como posibles mejoras, podríamos hacer un programa más complejo en Python o realizar otras tareas desde la terminal, como crear carpetas o manipular archivos, para entender aún más su gran capacidad de funcionamiento dentro del sistema operativo.

Las dificultades ya fueron detalladas en la sección de resultados. Pudimos resolverlas sin mayores problemas, ya que en la mayoría de los casos el mismo error indicaba cómo solucionarlo, o simplemente se trataba de errores de escritura en el código de Python.

7. Bibliografía

- Material teórico de la cursada “Arquitectura y Sistemas Operativos”. Tecnicatura Universitaria en Programación a Distancia – UTN FRBA. Año 2025.
- Oracle Corporation. (s.f.). *VirtualBox Downloads*. Recuperado el 1 de junio de 2025 de <https://www.virtualbox.org>
- Canonical. (s.f.). *Ubuntu Desktop Downloads*. Recuperado el 1 de junio de 2025 de <https://ubuntu.com/download/desktop>
- Lubuntu Team. (s.f.). *Lubuntu – Lightweight Linux*. Recuperado el 1 de junio de 2025 de <https://lubuntu.me>
- Odiard, C. A. (2023). *Video explicativo sobre Virtualización* [Video]. YouTube. <https://www.youtube.com/watch?v=Ne-IS7m9HEw>
- EmanuelABP. (2025). *Repositorio del Trabajo Práctico Final – Virtualización*. GitHub. <https://github.com/EmanuelABP/UTN-TP-Final-Virtualizacion>

8. Anexos

Se adjunta el siguiente repositorio de GitHub que contiene:

- Capturas del proceso de instalación y configuración de VirtualBox.
- Capturas del código ``ram.py`` ejecutado en la terminal.
- Archivos fuente del script y del informe.
- Registro de errores corregidos y pruebas realizadas.

Repositorio: <https://github.com/EmanuelABP/UTN-TP-Final-Virtualizacion>