

Trabajo Práctico Arquitectura y Sistemas Operativos

Virtualización

Alumnos:

Lautaro Lucero - lautalucero@gmail.com

Gabriela Machin - gabrielamachin.gm@gmail.com

Materia: Arquitectura y Sistemas Operativos

Profesores: David Roco y Ariel Enferrel

Fecha de Entrega: 05/06/2025



Índice:

- | | |
|-------------------------------|-------------|
| - Introducción ----- | pág. 3 |
| - Marco Teórico ----- | pág. 4 |
| - Caso Práctico ----- | pág. 6 |
| - Metodología Utilizada ----- | pág. 6 |
| - Resultados Obtenidos ----- | pág. 7 |
| - Conclusiones ----- | pág. 8 |
| - Bibliografía ----- | pág. 8 |
| - Anexos ----- | pág. 9 - 16 |

Introducción

La virtualización es una tecnología que permite crear representaciones digitales de componentes físicos como servidores, dispositivos de almacenamiento, redes y otros recursos informáticos. A través de software especializado, se simulan las funciones del hardware físico, lo que posibilita la ejecución simultánea de múltiples máquinas virtuales sobre un único equipo físico.

Elegimos este tema porque consideramos que representa una excelente oportunidad para ampliar nuestros conocimientos técnicos y aplicarlos en el contexto de nuestra formación como futuros técnicos en programación. Además, el dominio de herramientas de virtualización es altamente valorado en el ámbito profesional, especialmente en roles relacionados con el desarrollo, testing, administración de sistemas y despliegue de aplicaciones.

La virtualización no solo facilita la experimentación con distintos sistemas operativos, como Linux Mint o Ubuntu, sino que también permite crear entornos aislados para pruebas, simulaciones o capacitación, sin comprometer la integridad del sistema principal. Esta capacidad es útil tanto para nosotros como estudiantes y para profesionales que necesitan trabajar en entornos de desarrollo versátiles y seguros.

El objetivo de este trabajo es describir en detalle el funcionamiento de la virtualización, sus diferentes tipos y aplicaciones, y los escenarios en los que resulta más conveniente implementarla. Se incluirán ejemplos teóricos y un caso práctico, con el que buscamos demostrar nuestra comprensión de los conceptos fundamentales y la capacidad de aplicar herramientas de virtualización en contextos reales.

Marco Teórico

Como explica Stallings (2020), en los modelos tradicionales de computación, las aplicaciones se ejecutaban directamente sobre un sistema operativo instalado en una computadora personal o en un servidor. Esto implicaba que cada equipo sólo podía utilizar un sistema operativo a la vez, lo que generaba la necesidad de adaptar o modificar las aplicaciones para que funcionaran en cada sistema operativo o plataforma específica.

Ante este desafío surgió la virtualización como una solución eficiente. Esta tecnología permite que un solo equipo físico (ya sea una PC o un servidor) pueda operar en simultáneo varios sistemas operativos o múltiples instancias de un mismo sistema operativo. Gracias a la virtualización, es posible ejecutar diferentes aplicaciones, incluso aquellas diseñadas para plataformas distintas, dentro de un mismo entorno físico.

El sistema operativo anfitrión, mediante el uso de software especializado, puede crear y administrar varias máquinas virtuales (VM). Cada una de estas máquinas virtuales funciona como si fuera un equipo independiente, con su propio sistema operativo y, en algunos casos, incluso con una configuración de hardware virtualizada. Esto permite una mayor flexibilidad, aprovechamiento de recursos y compatibilidad en entornos heterogéneos.

A continuación, se describen los principales tipos de virtualización empleados en entornos informáticos:

Virtualización de servidores: Este tipo de virtualización consiste en particionar un único servidor físico en múltiples instancias virtuales independientes, conocidas como máquinas virtuales (VM). Cada VM puede ejecutar su propio sistema operativo y configurarse de forma autónoma, lo que permite maximizar el aprovechamiento de los recursos del hardware subyacente. Esta técnica contribuye a una mayor eficiencia operativa y a la consolidación de servidores.

Virtualización del almacenamiento: La virtualización del almacenamiento implica la unificación de varios dispositivos físicos de almacenamiento en una única entidad lógica. Mediante software especializado de gestión, estos recursos pueden asignarse y administrarse de manera dinámica según las necesidades de los

sistemas o aplicaciones, lo que favorece una mayor escalabilidad y optimización del uso del espacio disponible.

Virtualización de datos: En este caso, se introduce una capa de abstracción entre los datos físicos y las aplicaciones que los utilizan. Esta capa permite que las aplicaciones accedan a la información sin necesidad de conocer su ubicación exacta ni su formato, facilitando así una integración más ágil y personalizada de los datos según los requerimientos específicos de cada proceso o sistema.

Virtualización de redes: La virtualización de redes reemplaza funcionalidades que tradicionalmente eran desempeñadas por dispositivos físicos —como switches, routers o firewalls— mediante soluciones basadas en software. Esta transformación permite configurar, administrar y trasladar servicios de red con mayor flexibilidad, lo que reduce la dependencia del hardware específico y mejora la eficiencia operativa de la infraestructura de red.

De acuerdo con Silberschatz, Galvin y Gagne (2018), la virtualización de máquinas implica una serie de componentes clave. En el nivel más básico se encuentra el sistema físico, conocido como host o anfitrión, que proporciona la infraestructura de hardware sobre la cual se ejecutan las máquinas virtuales. Encima de este host opera el administrador de máquinas virtuales o hipervisor, cuya función principal es crear y gestionar las máquinas virtuales. Este componente ofrece a cada sistema invitado una interfaz que simula el entorno del host físico, lo que permite ejecutar distintos sistemas operativos de manera simultánea en un solo equipo, como si cada uno tuviera su propio hardware dedicado. En la mayoría de los casos, lo que se ejecuta dentro de cada máquina virtual es un sistema operativo completo.

Existen dos tipos principales de hipervisores. Los hipervisores de tipo 1, también llamados nativos o de bare-metal, se ejecutan directamente sobre el hardware físico sin necesidad de un sistema operativo intermedio. Entre los ejemplos más representativos de esta categoría se encuentran VMware ESX, Joyent SmartOS y Citrix XenServer.

Por otro lado, los hipervisores de tipo 2 operan como aplicaciones dentro de un sistema operativo convencional, proporcionando capacidades de virtualización

desde ese entorno. Algunos ejemplos comunes incluyen VMware Workstation, Parallels Desktop y Oracle VirtualBox.

Además de estas formas de virtualización, existen soluciones que, si bien no virtualizan sistemas completos, ofrecen un aislamiento similar para las aplicaciones. Estas tecnologías, como Oracle Solaris Zones, BSD Jails o IBM AIX WPARs, permiten contener aplicaciones dentro de entornos controlados, incrementando la seguridad y facilitando la administración.

En el contexto del desarrollo de software, la virtualización permite a los desarrolladores crear y utilizar máquinas virtuales para aislar los entornos de desarrollo y prueba, evitando interferencias entre proyectos y garantizando condiciones controladas para cada etapa del ciclo de vida del software. Esto mejora la eficiencia en el uso de los recursos del sistema y refuerza la seguridad, al minimizar los riesgos asociados a la ejecución de código no verificado o potencialmente dañino en entornos de producción. Además, la virtualización permite replicar fácilmente configuraciones específicas, facilitando la detección y resolución de errores en condiciones similares a las del entorno final del usuario.

Caso Práctico

Se utilizó VirtualBox para instalar una máquina virtual con Linux Mint y se realizó la instalación de Python (3.12).

- Crear una nueva máquina virtual con 4 GB de RAM, 4 hilos del procesador y 50 GB de almacenamiento.
- Montar ISO de Linux Mint y realizar instalación mínima del mismo.
- Instalar Python y probar el programa con un código simple.

Metodología Utilizada

A continuación, se detallan los pasos llevados a cabo durante el desarrollo del trabajo práctico:

1. Instalación de VirtualBox: Se descargó e instaló el software de virtualización VirtualBox desde su sitio oficial en un sistema operativo Windows.
2. Descarga de la imagen ISO de Linux Mint: Se obtuvo la imagen ISO oficial de Linux Mint (edición Cinnamon) desde el sitio web del proyecto.
3. Creación de la máquina virtual: Se procedió a crear una nueva máquina virtual en VirtualBox, asignándole recursos específicos de hardware (memoria RAM, espacio en disco, número de procesadores) y configuraciones básicas de software.
4. Instalación del sistema operativo: Se instaló Linux Mint en la máquina virtual utilizando la ISO previamente descargada.
5. Actualización del sistema e instalación de Python: Desde la terminal del sistema operativo virtualizado, se ejecutaron los siguientes comandos:
`sudo apt update`
`sudo apt install python3`
6. Verificación de la instalación de Python: Para comprobar que Python 3.12 se instaló correctamente, se utilizó el comando:
`python3 --version`
7. Prueba de ejecución en Python: Se accedió al intérprete de Python mediante el comando `python3`, y se ejecutó un pequeño script para verificar su correcto funcionamiento:
`print("Hello, World!")`
8. Instalación de Visual Studio Code: Finalmente, se instaló el entorno de desarrollo Visual Studio Code, el cual fue utilizado para editar y probar un script más complejo, correspondiente a una calculadora.

Resultados Obtenidos

La máquina virtual se ejecutó correctamente, permitiendo el funcionamiento estable del entorno virtualizado. Se logró instalar exitosamente Python 3.12, el cual fue probado y funcionó sin inconvenientes. Asimismo, se completó de manera satisfactoria la instalación del entorno de desarrollo Visual Studio Code.

Durante la ejecución de un ejemplo práctico (una calculadora en Python), inicialmente no se pudo ejecutar el programa debido a la falta de ciertas librerías necesarias. Sin embargo, el inconveniente fue identificado rápidamente y solucionado mediante la instalación correspondiente de los módulos requeridos. Las capturas de pantalla que documentan este proceso se incluyen en la sección de Anexos.

Conclusiones

Durante el desarrollo del trabajo, aprendimos a utilizar máquinas virtuales para ejecutar un entorno Linux sobre un sistema operativo Windows, sin necesidad de desinstalar este último. Para ello, empleamos la herramienta VirtualBox, que nos permitió crear y configurar una máquina virtual de manera eficiente.

Uno de los principales desafíos que encontramos durante la realización del trabajo fue la activación de la virtualización desde la BIOS del equipo. En nuestro caso, la opción se encontraba en un apartado distinto al indicado durante las clases teóricas de la asignatura, lo que requirió realizar una búsqueda adicional por nuestra cuenta. Además, cada integrante del grupo contaba con un equipo con diferentes características de hardware, por lo que fue necesario coordinar e investigar en conjunto para establecer un entorno de virtualización homogéneo que cumpliera con los requisitos del trabajo.

En futuras instancias, consideramos que sería enriquecedor explorar otras soluciones de virtualización como VMware Workstation, Parallels Desktop o QEMU, así como experimentar con distintas distribuciones de Linux (por ejemplo, Ubuntu, Lubuntu o Manjaro) o incluso implementar servicios más complejos, como la

instalación y configuración de un servidor web Apache, con el objetivo de ampliar nuestros conocimientos y habilidades prácticas en entornos virtualizados.

Bibliografía

- Stallings, W. - Operating Systems: Internals and Design Principles, Pearson, 2020.
- Silberschatz, A., Galvin, P. B., Gagne, G. - Operating System Concepts, Wiley, 2018.
- Documentación oficial de VirtualBox: <https://www.virtualbox.org/manual/>
- Documentación oficial de Linux Mint: <https://linuxmint.com/documentation.php>
- Conceptos sobre la virtualización en sitio oficial de AWS:
<https://aws.amazon.com/es/what-is/virtualization/>

Anexos

A continuación, se adjuntan las capturas de pantalla a modo de evidencia del proceso de instalación y configuración del entorno.

Imagen 1: Luego de la instalación de VirtualBox, se comienza la creación de la máquina virtual.

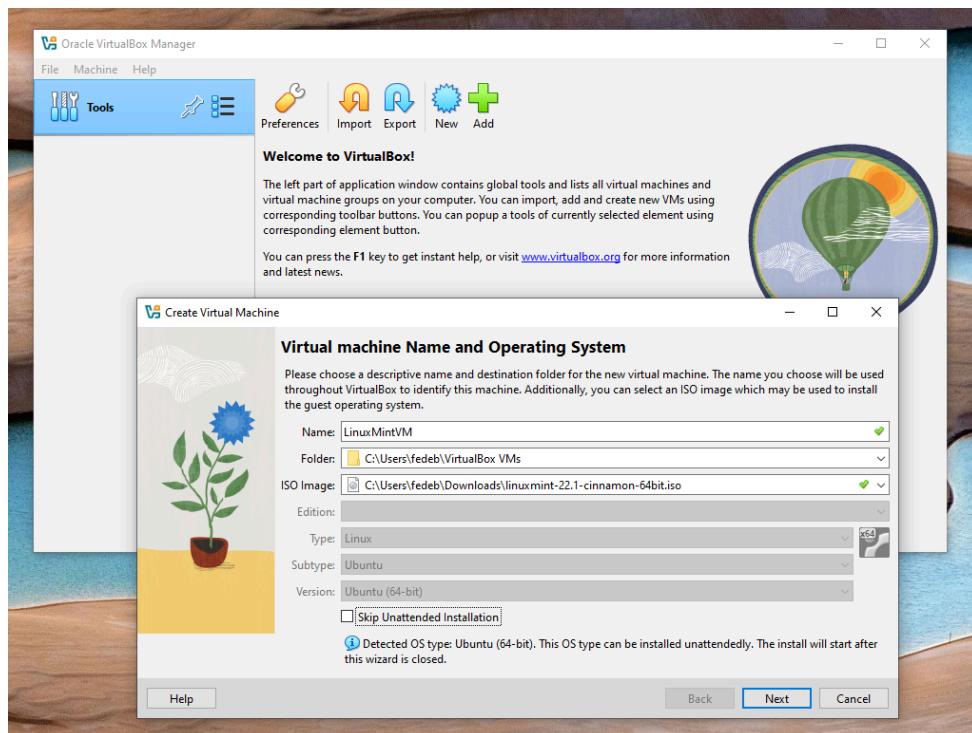


Imagen 2: Configuración de usuario y contraseña para reforzar la seguridad de la VM.

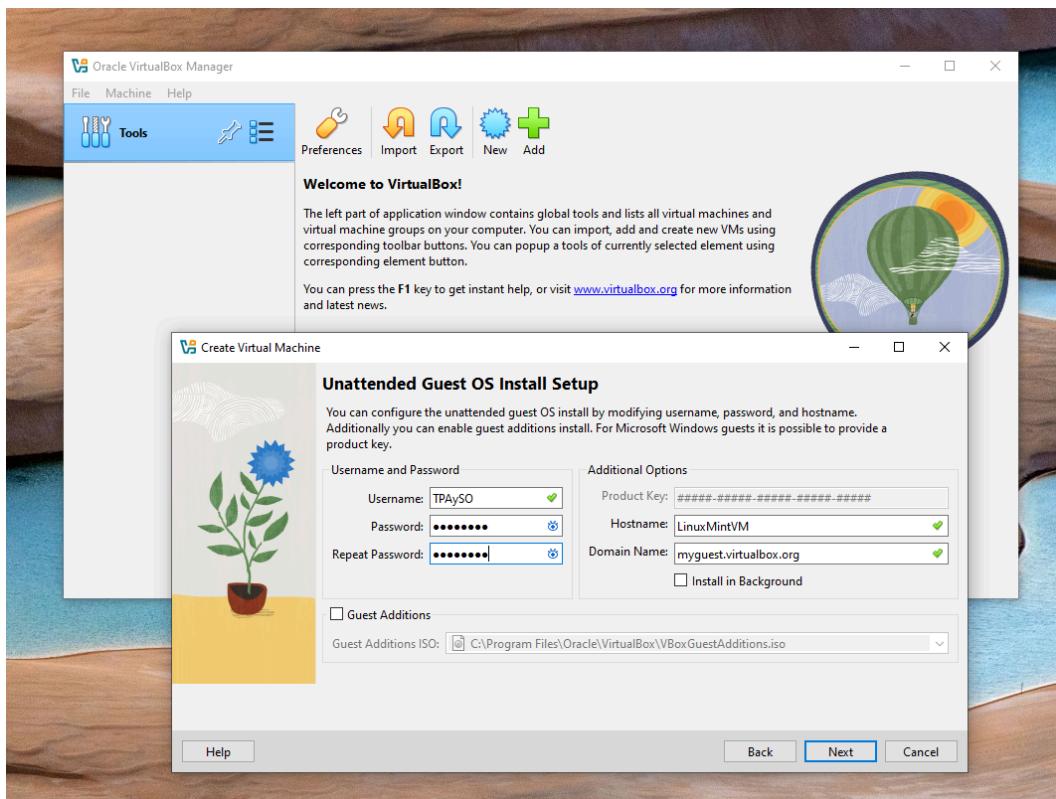


Imagen 3: Configuración del hardware

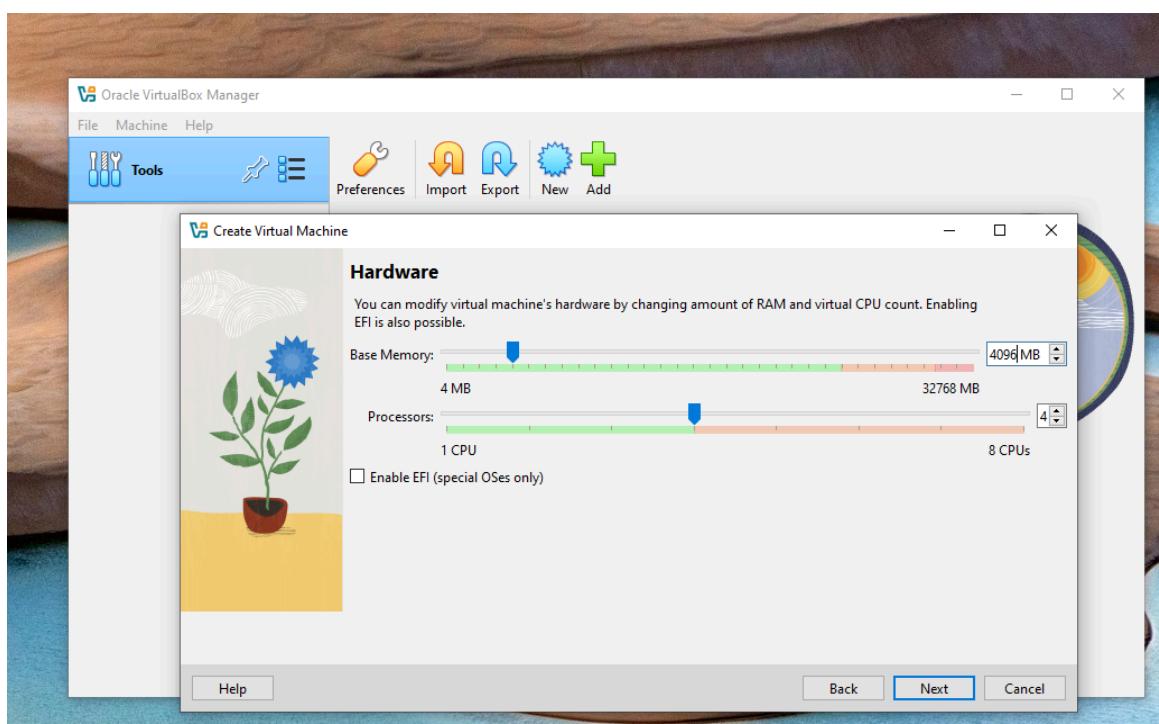


Imagen 4: Configuración del almacenamiento

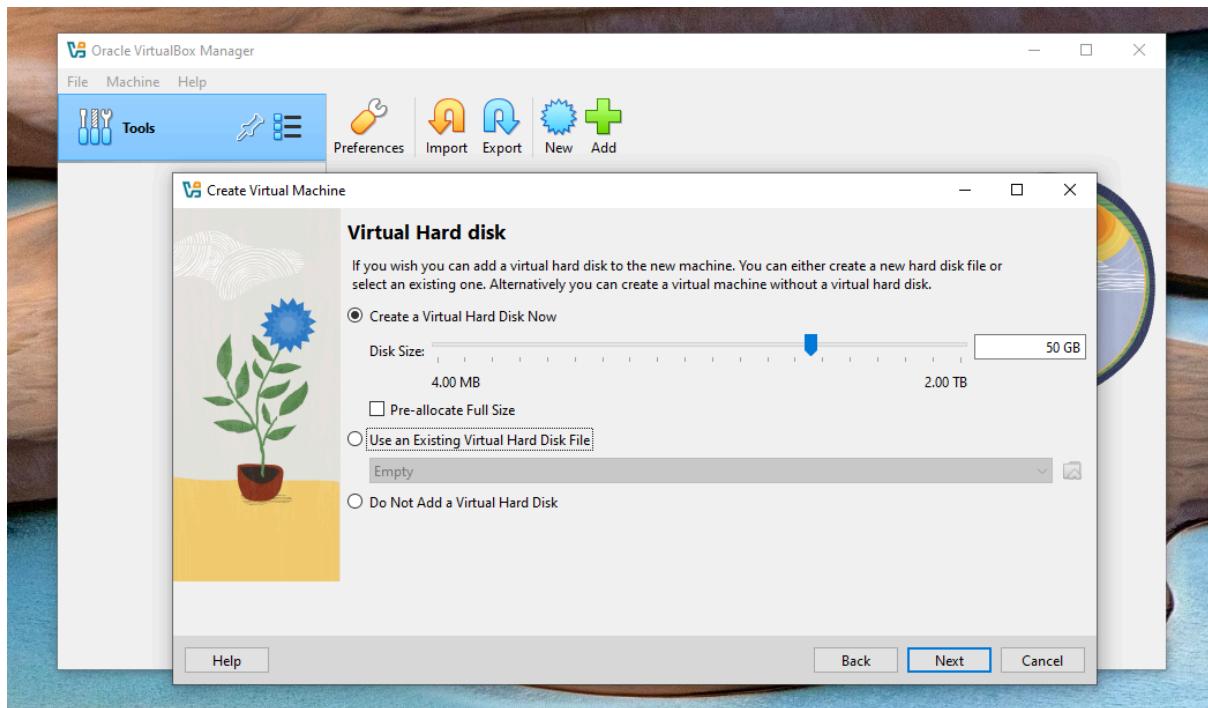


Imagen 5: Resumen de la configuración de la VM

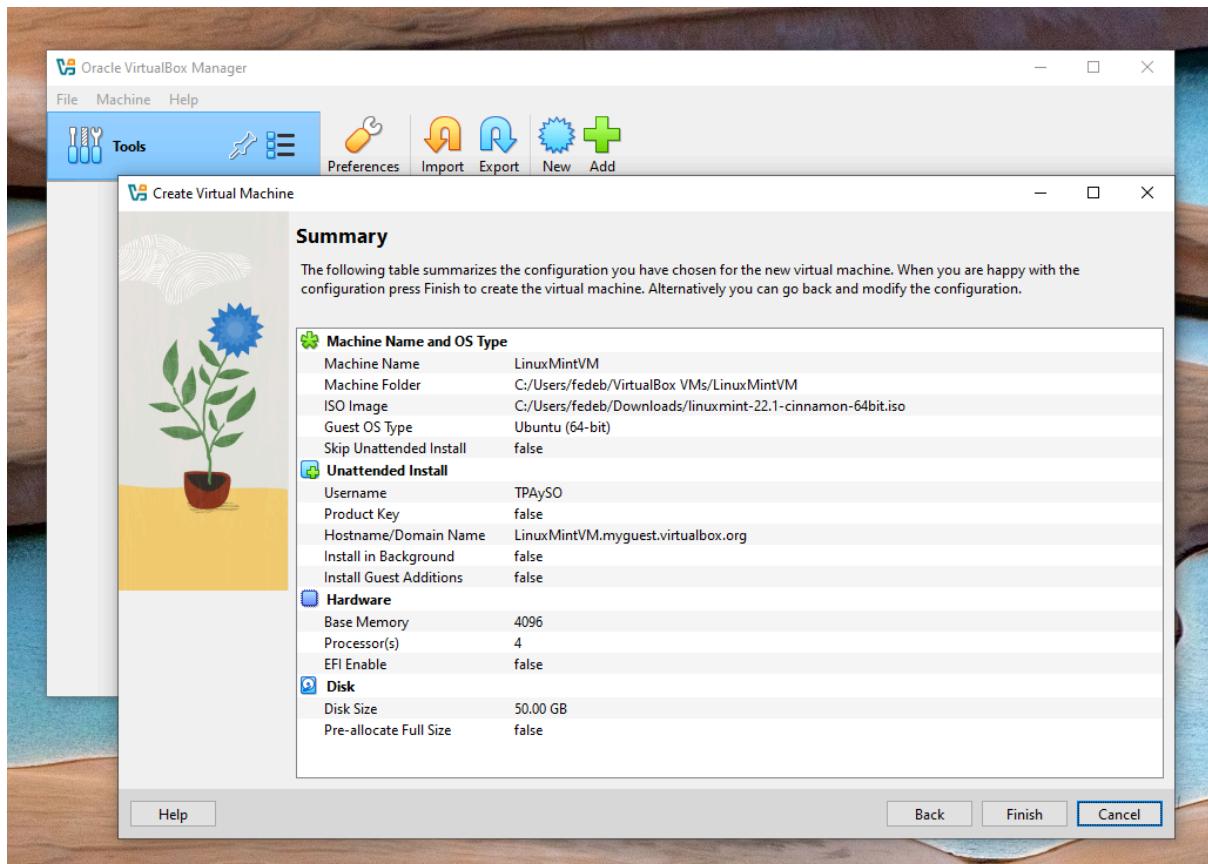


Imagen 6: Finalización del proceso de creación (VM apagada)



Imagen 7: Inicialización de la VM por primera vez

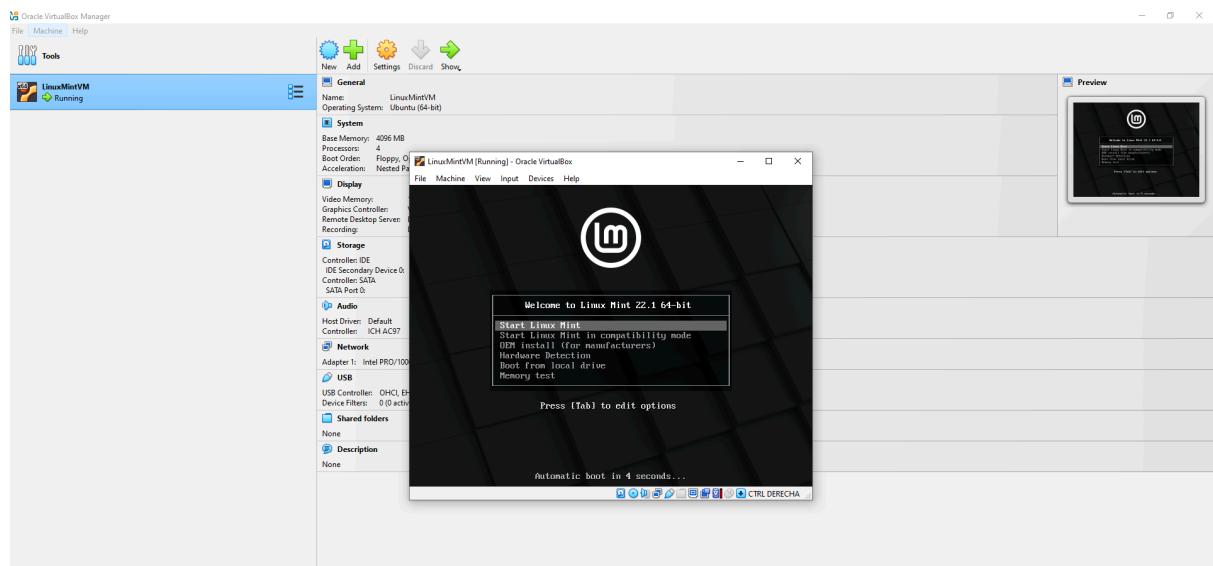


Imagen 8: VM funcionando antes de la instalación de Linux Mint

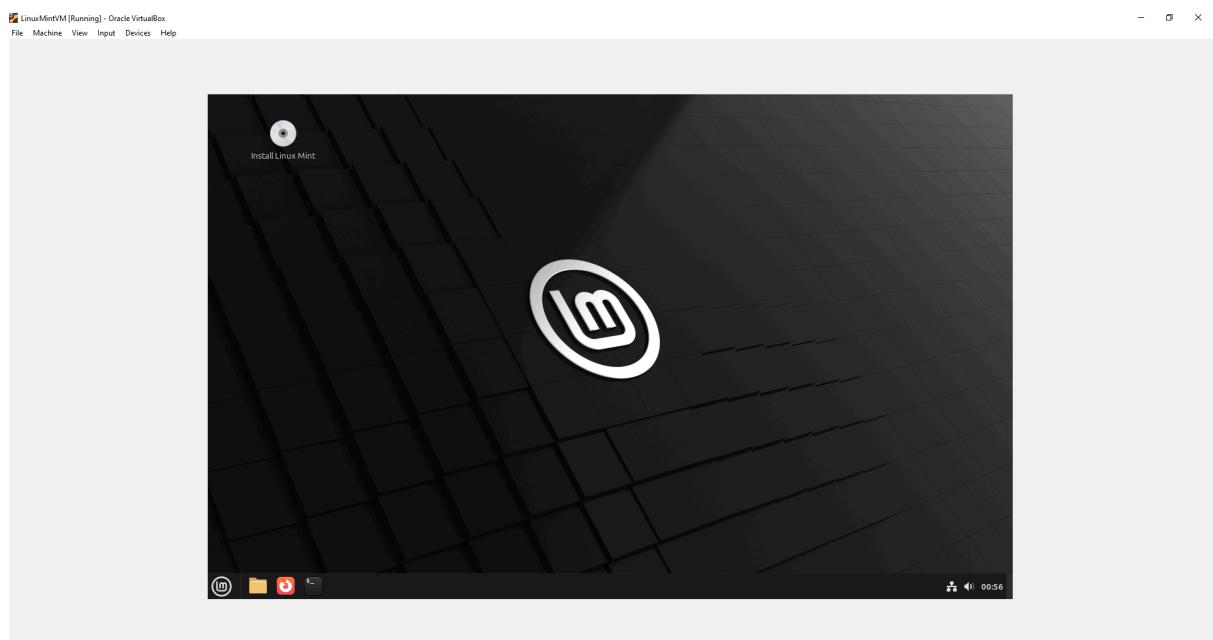


Imagen 9: Proceso de instalación del SO elegido dentro de la VM

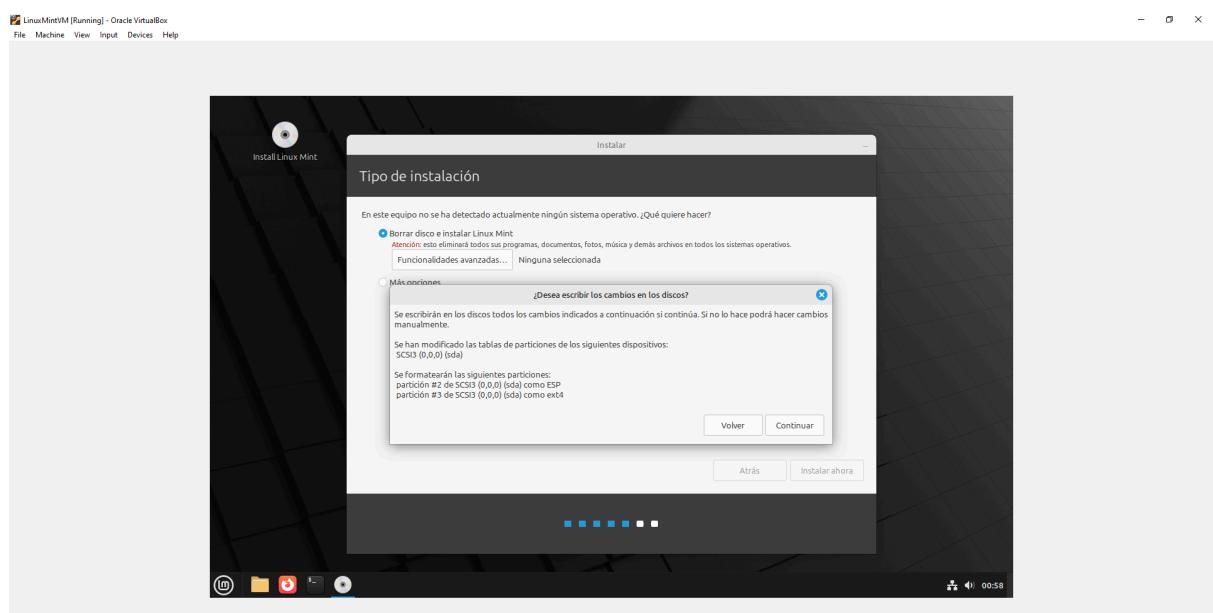


Imagen 10: Configuración de usuario y contraseña, para reforzar la seguridad al acceder al sistema

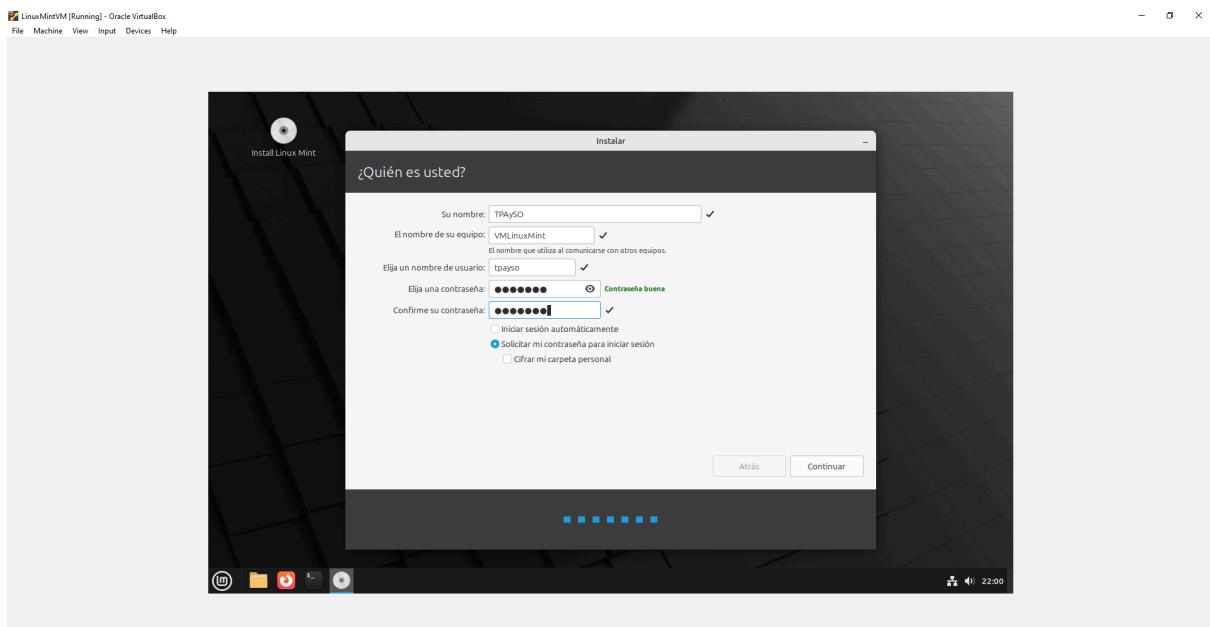


Imagen 11: Se inicia sesión una vez finalizado el proceso de instalación

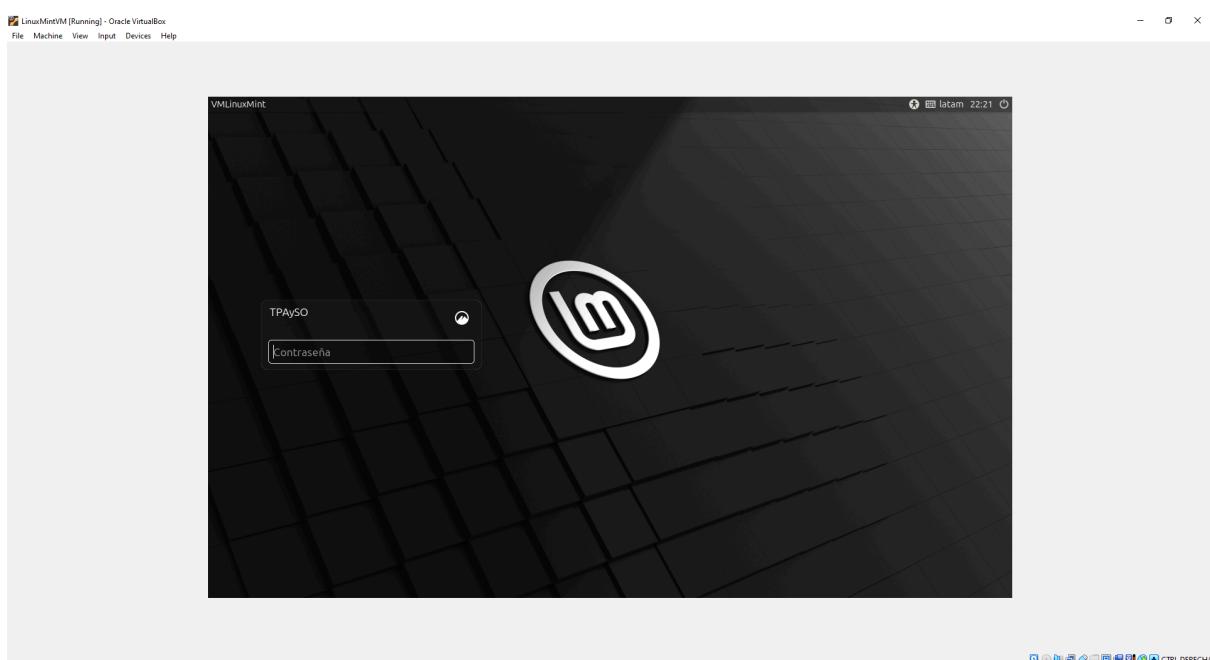


Imagen 12: Acceso exitoso al sistema

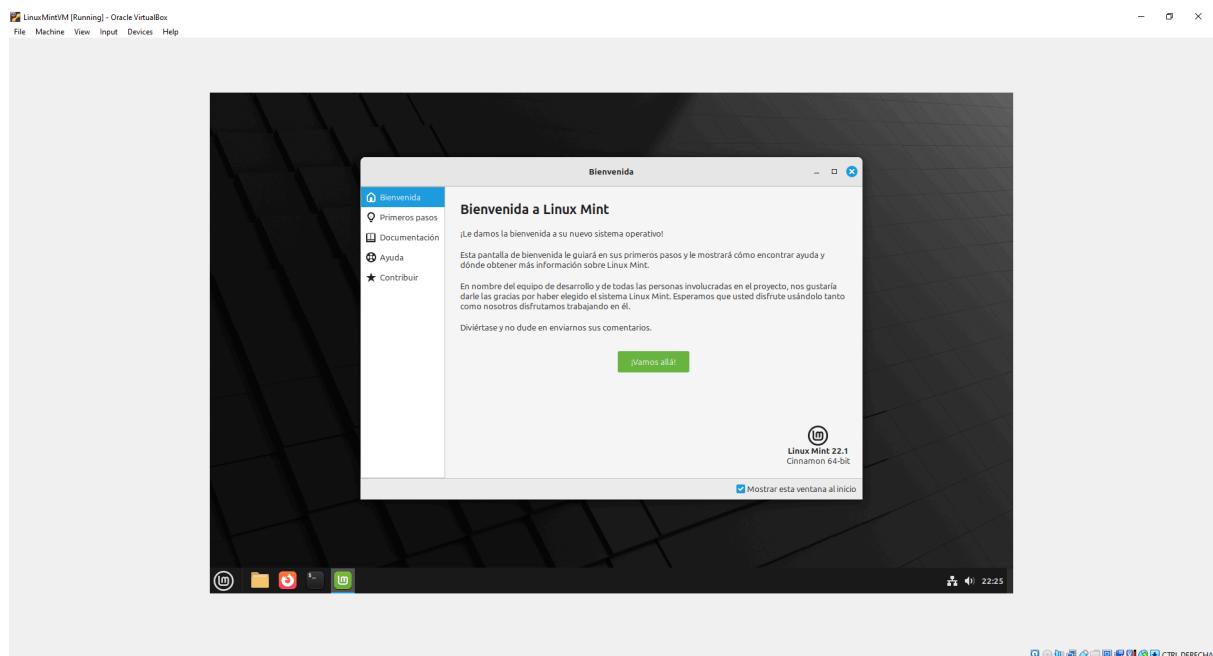


Imagen 13: Ejecución de comandos en la terminal de Linux Mint en la VM

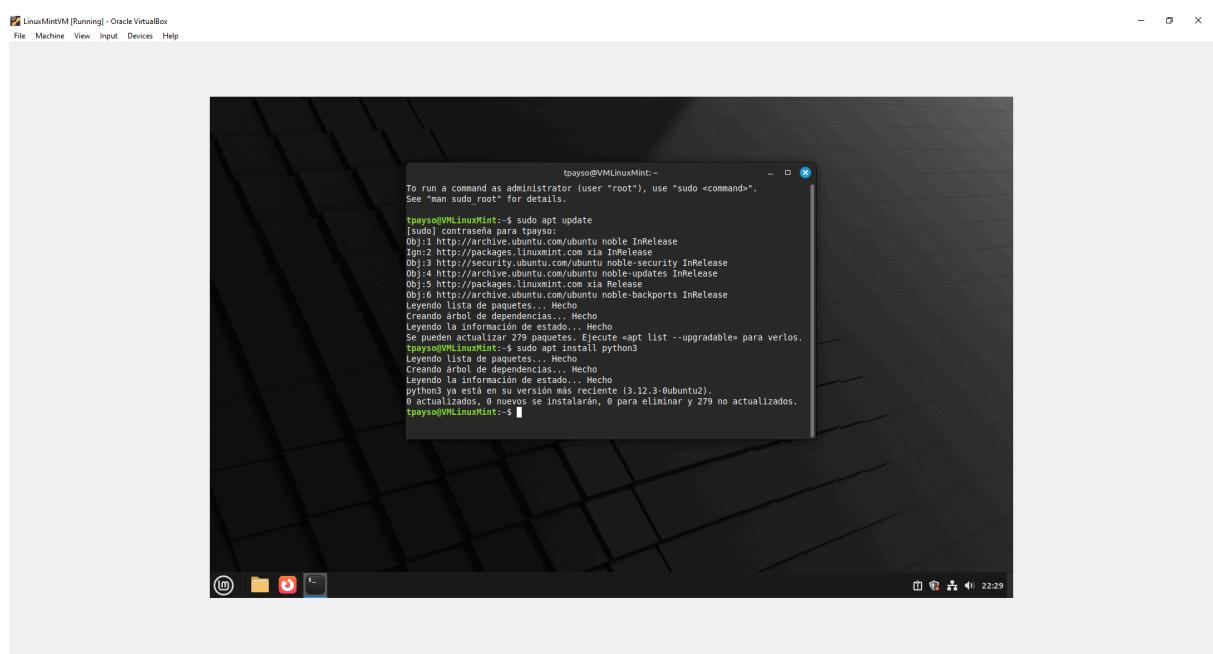


Imagen 14: Verificación de la versión de Python y ejecución de un breve script para verificar que funciona correctamente

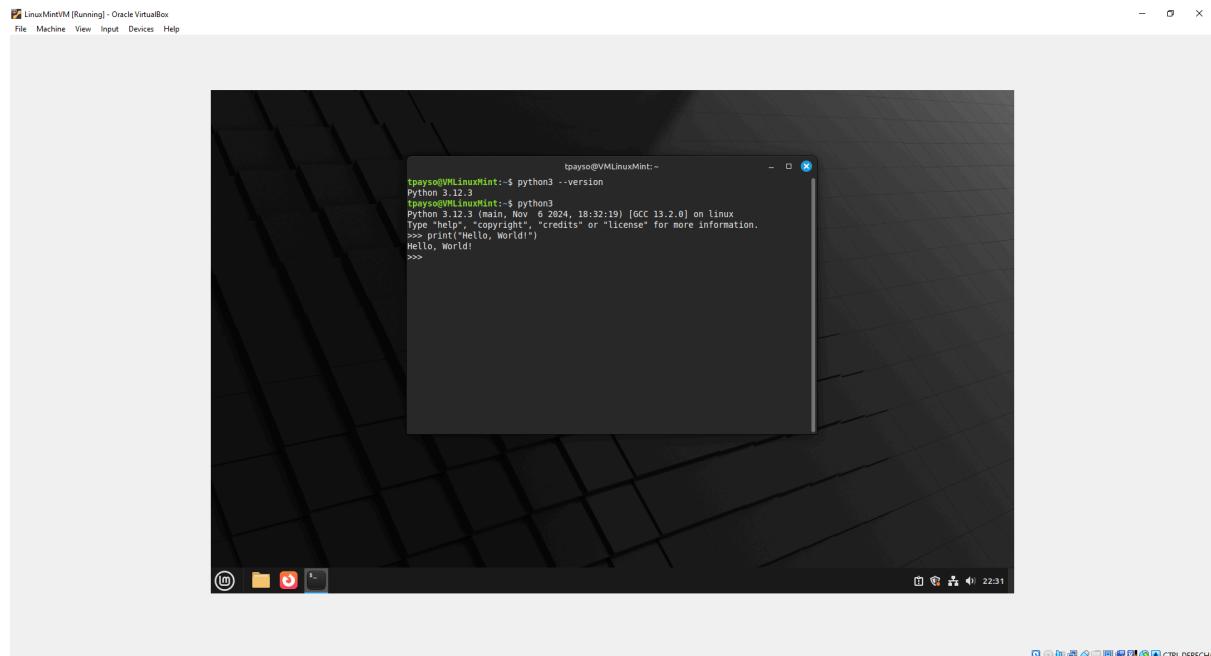


Imagen 15: Código Python del proyecto en Visual Studio Code en el entorno Linux generado en la máquina virtual

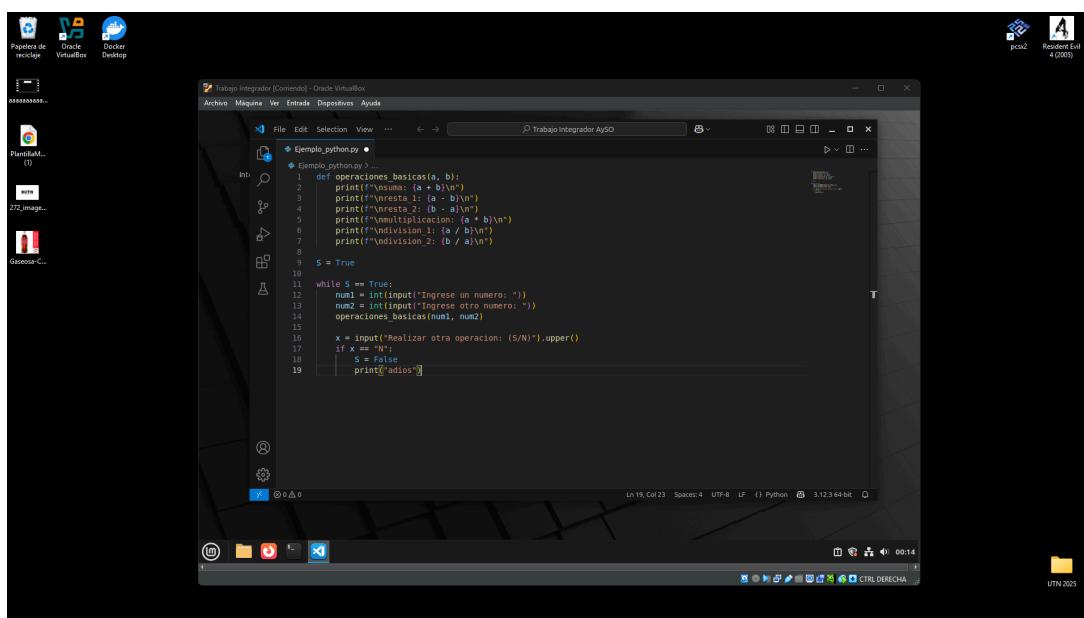


Imagen 16: Prueba del código

The screenshot shows a Linux desktop environment with a dark theme. A terminal window titled "Trabajo Integrador [Comiendo] - Oracle VM VirtualBox" is open, displaying Python code and its execution. The code defines a function to calculate the difference between two numbers and then calls it with user input. The terminal output shows the result and asks if the user wants to perform another operation.

```
def operaciones_basicas(a, b):
    print("\nresta 1: (a - b)\n")
    print("resta 2: (b - a)\n")
    print("multiplicacion: (a * b)\n")
    print("division 1: (a / b)\n")
    print("division 2: (b / a)\n")

Realizar otra operación: (S/N)[]
```

Ln 17, Col 17 Spaces: 4 UTF-8 LF Python 3.12.3 64-bit

- Link del video en Google Drive: [0605.mp4](#)
- Link del repositorio en GitHub:
<https://github.com/lautarillo/TRABAJO-INTEGRADOR-AySO-LUCERO-Y-MACHIN/tree/master>