

# **Trabajo Final Integrador: Virtualización**

## **Alumnos:**

**Benjamín Ugarte - benjamintobiasugarte2003@gmail.com - Comisión 22**

**Nicolás Colman - nicolas.colman@tupad.utn.edu.ar - Comisión 11**

## **Materia:**

**Arquitectura y Sistemas Operativos**

## **Profesores:**

**Osvaldo Falabella**

**Diego Lobos**

**Fecha de entrega: 5/6/2025**



## Índice

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

## **1. Introducción**

La virtualización es una tecnología que permite ejecutar uno o más sistemas operativos dentro de otro sistema, como si se tratara de computadoras independientes, aunque en realidad todo funciona dentro de una misma PC. Esto es posible gracias a un software específico que actúa como intermediario entre el hardware real y los sistemas virtuales, permitiendo simular recursos como memoria, almacenamiento, red y procesador. Mediante esta herramienta es posible, por ejemplo, tener una computadora con Windows y al mismo tiempo ejecutar una distribución de Linux dentro de una máquina virtual, sin necesidad de modificar el sistema principal.

Se eligió el tema de virtualización porque, aunque se tenía una noción básica por el uso de aplicaciones como BlueStacks –que permite emular Android- no se conocía en profundidad el uso de VirtualBox. Esta curiosidad despertó el interés por aprender cómo funciona esta tecnología y cuáles son sus aplicaciones concretas en el entorno técnico y profesional. En el presente trabajo integrador se detalla el proceso de instalación y configuración de VirtualBox, un hipervisor de tipo 2, así como la creación de una máquina virtual que corre el sistema operativo Ubuntu.

La virtualización tiene importancia en la formación como técnico en programación ya que permite acceder y experimentar con otros sistemas operativos, más allá de Windows o Mac OS. Sistemas como Linux, Ubuntu o CentOS son ampliamente utilizados en entornos de desarrollo y servidores. Poder crear y gestionar máquinas virtuales con estos sistemas permite realizar pruebas de compatibilidad, simular entornos de producción y entender cómo se comportan las aplicaciones en distintos contextos. Con el desarrollo de este trabajo se busca comprender el concepto de virtualización y su funcionamiento. Instalar y configurar máquinas virtuales utilizando VirtualBox. Adquirir conocimientos sobre otros sistemas operativos y su entorno de uso. Analizar las principales ventajas y desventajas de la virtualización.

## **2. Marco Teórico**

### **1. Concepto de virtualización**

La virtualización es una técnica informática que permite crear versiones virtuales de recursos físicos, como servidores, sistemas operativos, redes o discos duros. Según Stallings (2012), consiste en separar el software del hardware para permitir múltiples

entornos operativos de manera independiente dentro de una misma maquina física. Esta abstracción mejora el aprovechamiento de los recursos disponibles y permite mayor flexibilidad en la gestión de entornos de trabajo.

## **2. Virtualización de sistemas operativos**

Existen distintas formas de virtualización, siendo la virtualización de sistemas operativos una de las más comunes. En esta, se utiliza un hipervisor para crear entornos independientes que simulan computadoras completas. A través de herramientas como VirtualBox, VMware o Hyper-V los usuarios pueden crear y administrar entornos virtuales de forma sencilla, aislada y segura. Esta virtualización resulta especialmente útil en entornos educativos, desarrollo y de prueba.

## **3. Hipervisores**

Los hipervisores se clasifican principalmente en dos tipos:

- Hipervisor de tipo 1: se ejecuta directamente sobre el hardware físico y es utilizado principalmente en entornos empresariales y servidores.
- Hipervisor de tipo 2: funciona sobre un sistema operativo ya instalado, actuando como una aplicación más. Es ideal para entornos de escritorio o educativos, como es el caso de VirtualBox, que permite ejecutar, por ejemplo, sistemas como Linux dentro de una máquina con Windows.

## **4. Beneficios de la virtualización**

La virtualización ofrece una serie de beneficios que la han convertido en una tecnología ampliamente utilizada:

- Aislamiento: cada máquina virtual funciona de forma independiente, lo que permite ejecutar pruebas sin riesgo de afectar el sistema principal.
- Ahorro de recursos: se evita la necesidad de tener multiples equipos físicos para probar distintos entornos.
- Flexibilidad: se pueden administrar máquinas virtuales fácilmente.

- Seguridad: cualquier falla dentro de una máquina virtual no compromete el sistema operativo anfitrión.

## **5. Ubuntu como sistema operativo virtualizado**

Ubuntu es una de las distribuciones más populares del sistema operativo Linux. Se usa para prácticas de administración de servidores, redes y desarrollo en entornos virtualizados, por su estabilidad y comunidad activa. Al ejecutarse dentro de una máquina virtual, Ubuntu permite simular entornos de servidores reales, realizar pruebas de software, explorar comandos del sistema y comprender la arquitectura de sistemas basados en Unix, sin necesidad de alterar el sistema operativo principal del equipo.

## **6. Visual Studio Code**

Visual Studio Code es un editor de código fuente avanzado y multiplataforma, desarrollado por Microsoft. Fue utilizado en este trabajo para editar y gestionar scripts en el entorno virtualizado. Esta herramienta nos permite trabajar con distintos lenguajes de programación, ofrece funcionalidades como autocompletado, resaltado de sintaxis, terminal integrada y extensiones útiles para entornos Linux, lo cual facilita la interacción con sistemas operativos virtualizados como Ubuntu.

## **7. Git y GitHub**

Git es un sistema de control de versiones distribuido que permite gestionar cambios en el código fuente de manera eficiente y segura. Por su parte, GitHub es una plataforma basada en Git que se utilizó para almacenar y documentar el proyecto integrador. Gracias a GitHub se pudo organizar el código, subir evidencias (como capturas y código fuente), y facilitar la colaboración entre los integrantes del grupo.

## Caso Práctico

Se utilizó VirtualBox para instalar una máquina virtual con Ubuntu Server 24.04, además se utilizó VSCode para desarrollar un programa en python.

Pasos:

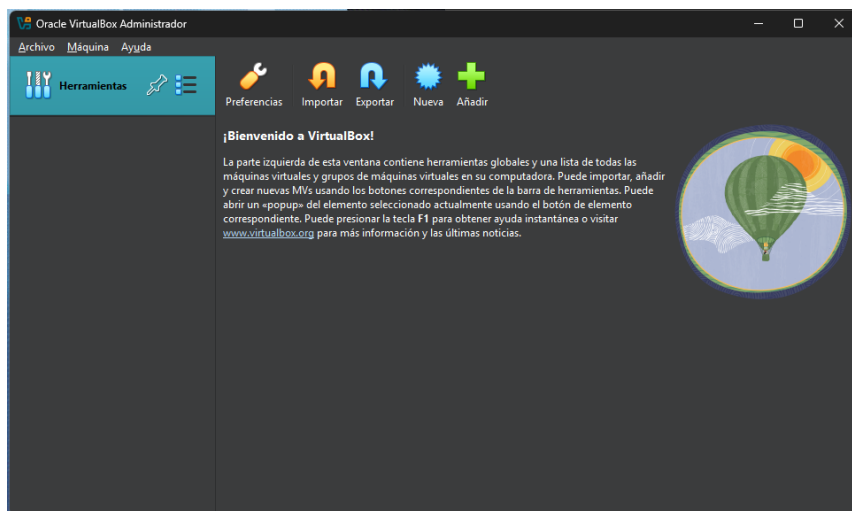
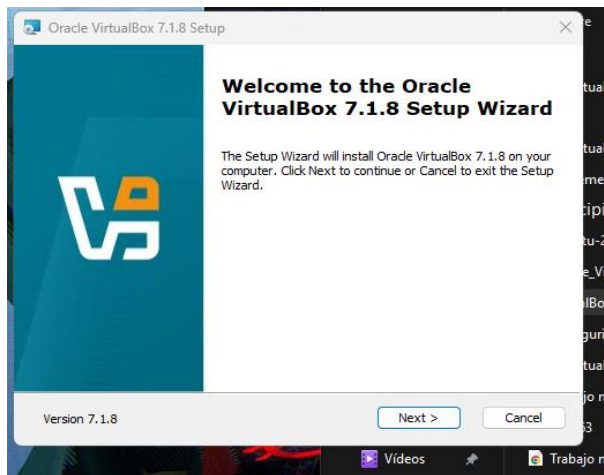
### Primer paso: Descarga de VirtualBox

Como primer paso, se descargó el programa VirtualBox desde su sitio oficial:

<https://www.virtualbox.org>. También se descargó la imagen ISO de Ubuntu en su versión 24.04.2, desde la página oficial del sistema operativo: <https://ubuntu.com/download/desktop>

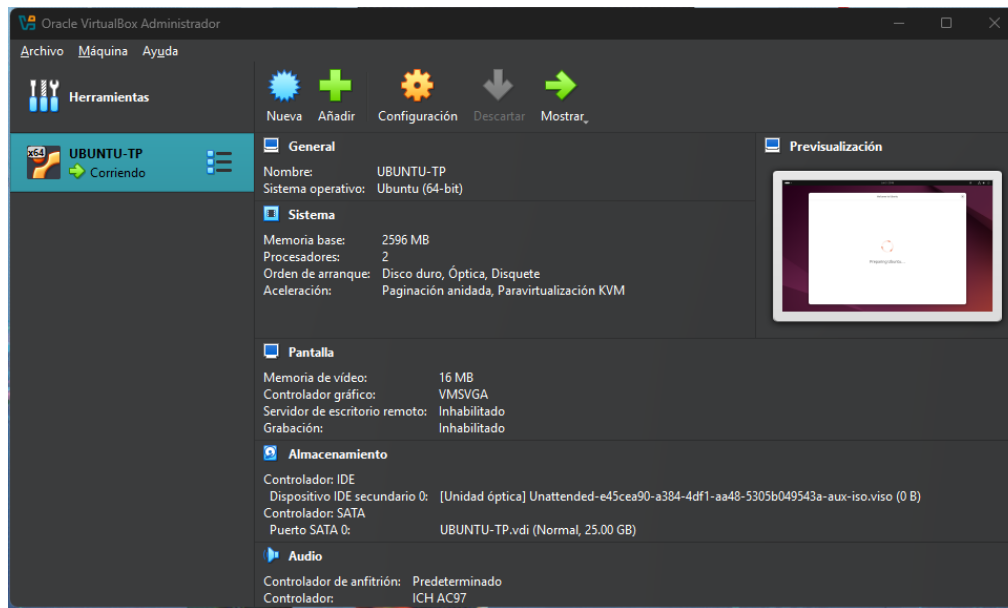
### Segundo paso: Instalación de VirtualBox

Una vez obtenidos los archivos, se ejecutó el instalador de VirtualBox y se realizó la instalación en el sistema operativo Windows siguiendo el asistente paso a paso. Este proceso no presentó inconvenientes y dejó el programa listo para crear una máquina virtual.



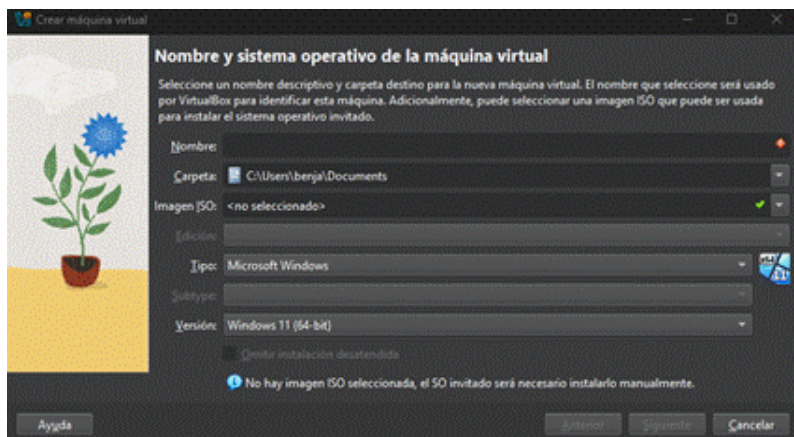
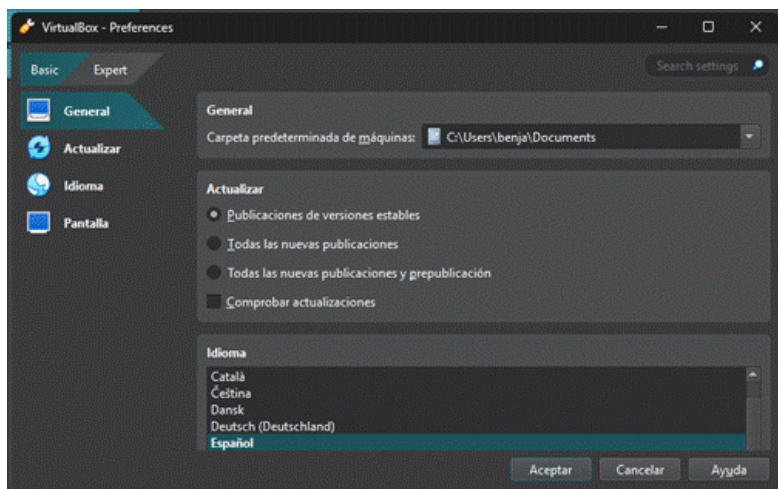
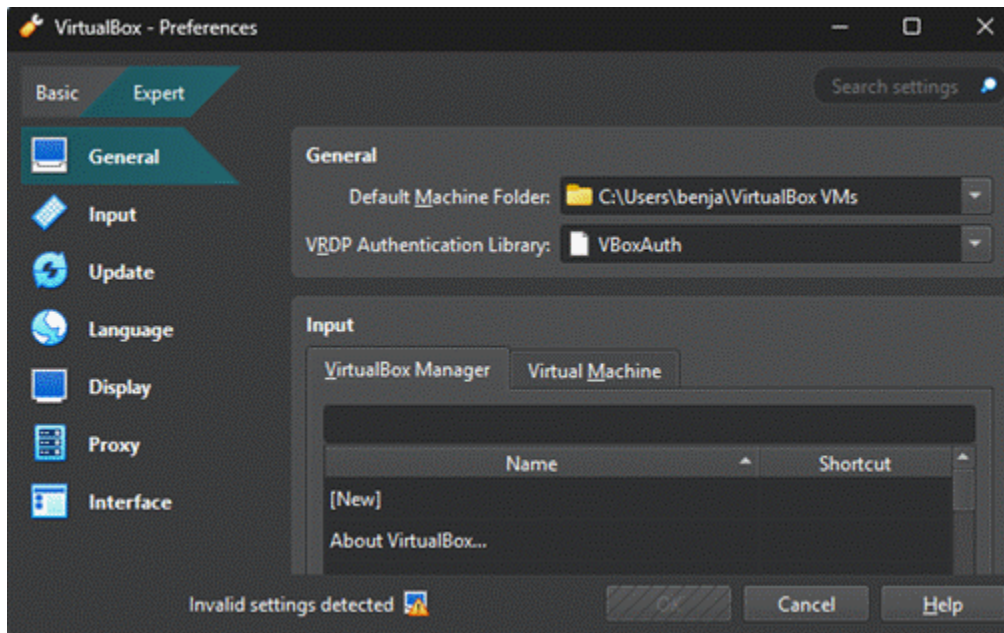
### Tercer paso: Creación de máquina virtual

- Se procede a crear la máquina virtual con Ubuntu
- Se eligió el nombre UBUNTU-TP y el archivo ISO de Ubuntu
- Se cambio el nombre de usuario y la contraseña
- Se eligió que cantidad de RAM, núcleos y almacenamiento
  - RAM 2590 MB
  - Núcleos 2
  - Almacenamiento 25.0 GB

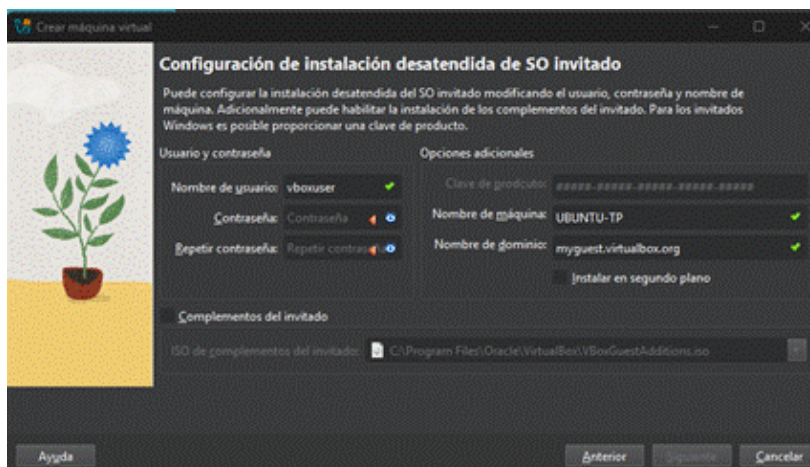
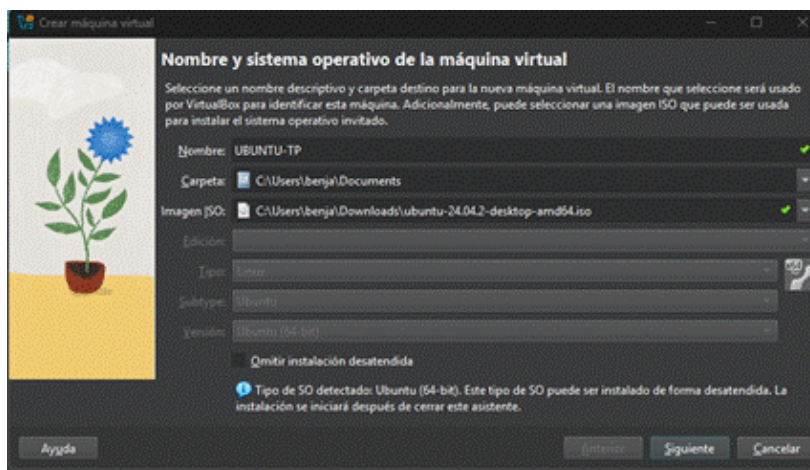
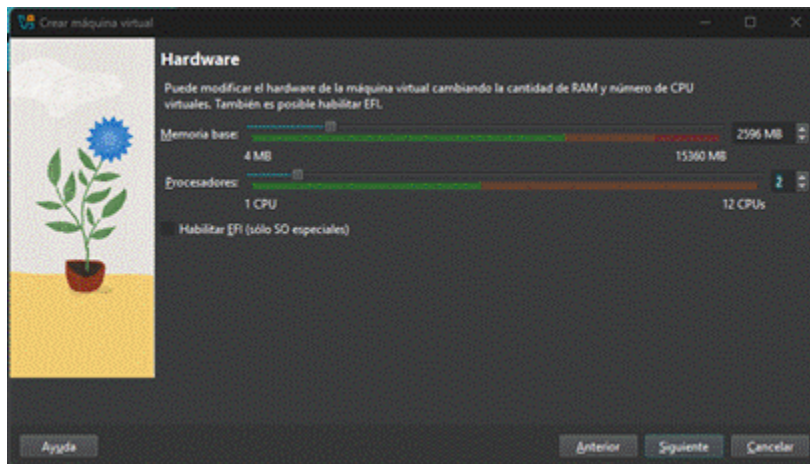


En el caso de Benjamin Ugarte tuvo un inconveniente con la creación

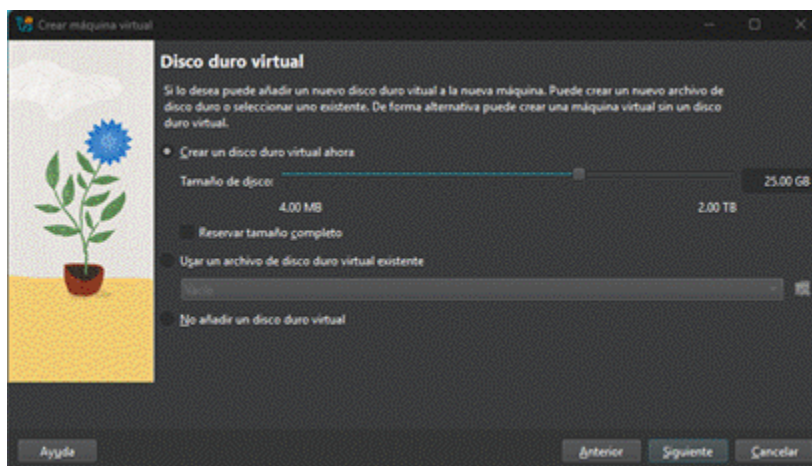
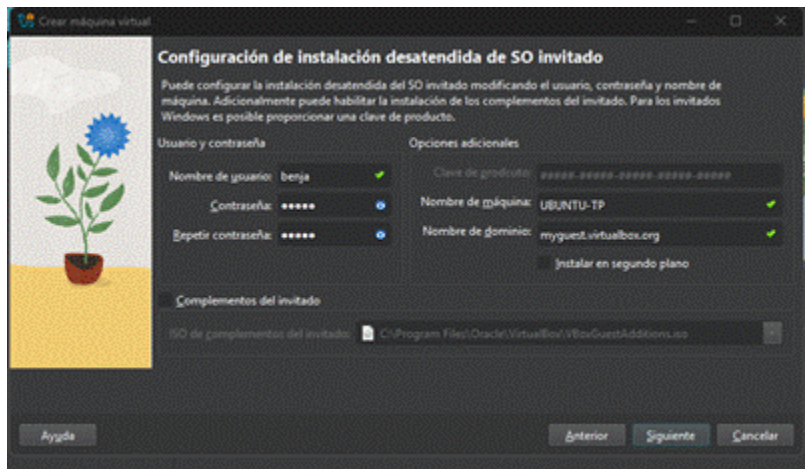




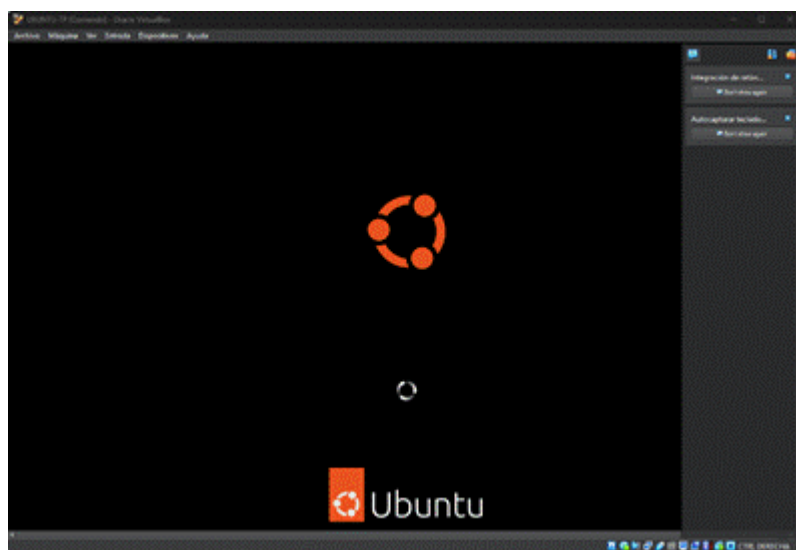
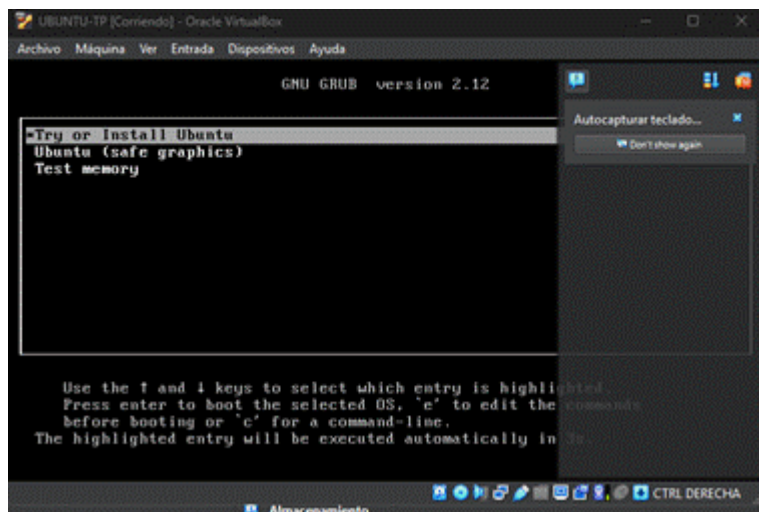
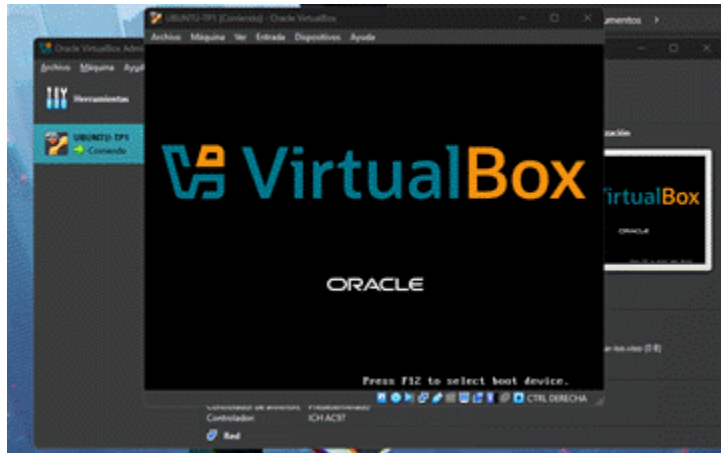
# TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA



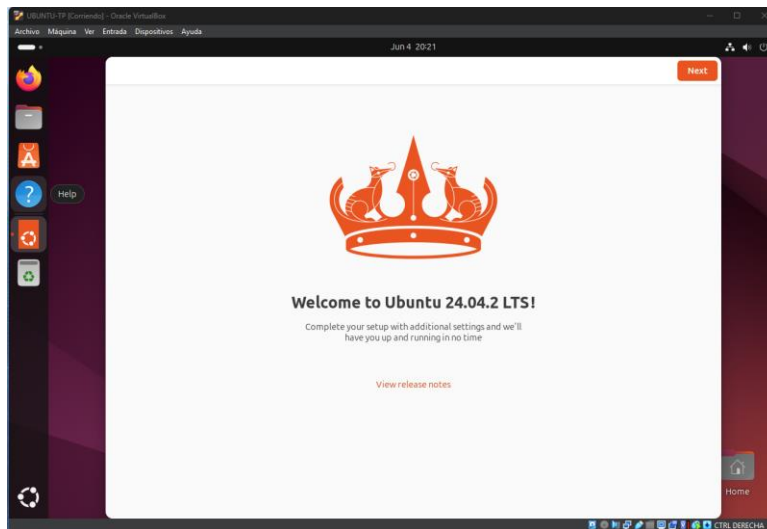
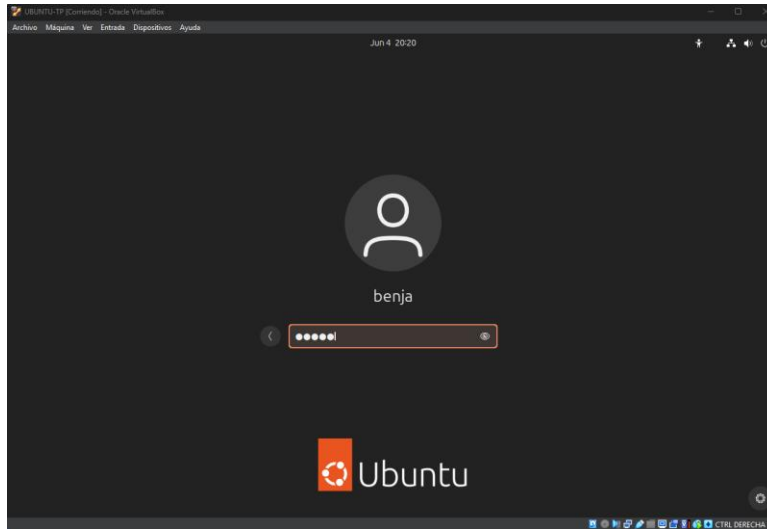




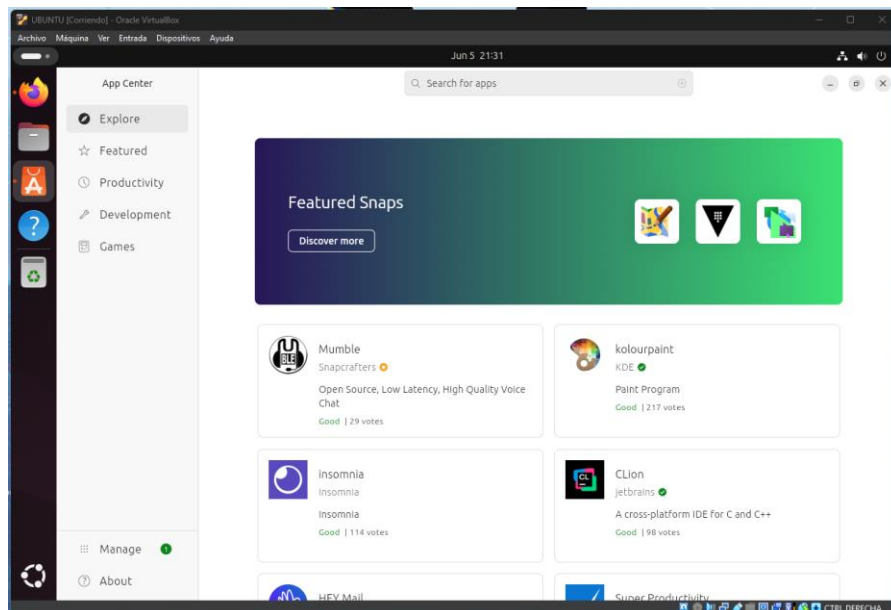
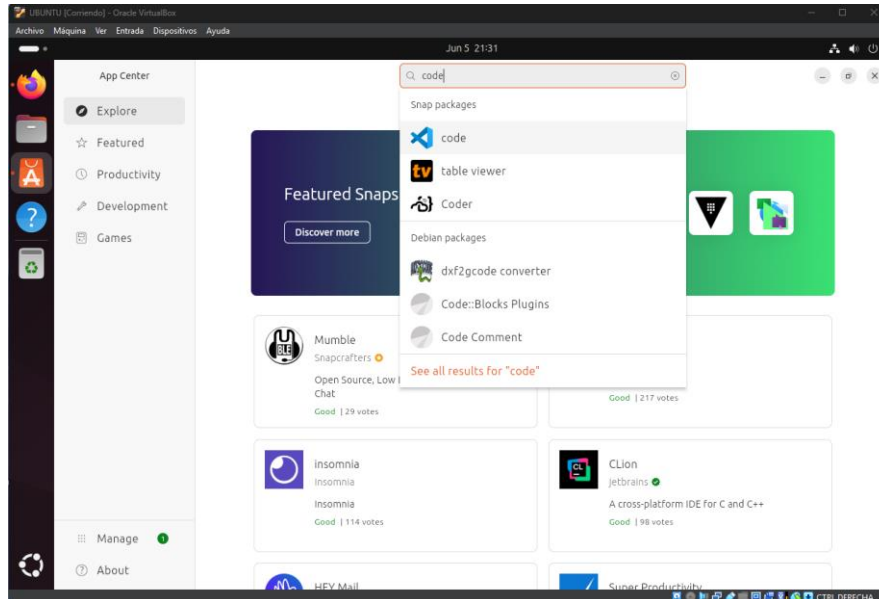
Cuarto paso: se procedió a instalar Ubuntu en el VM



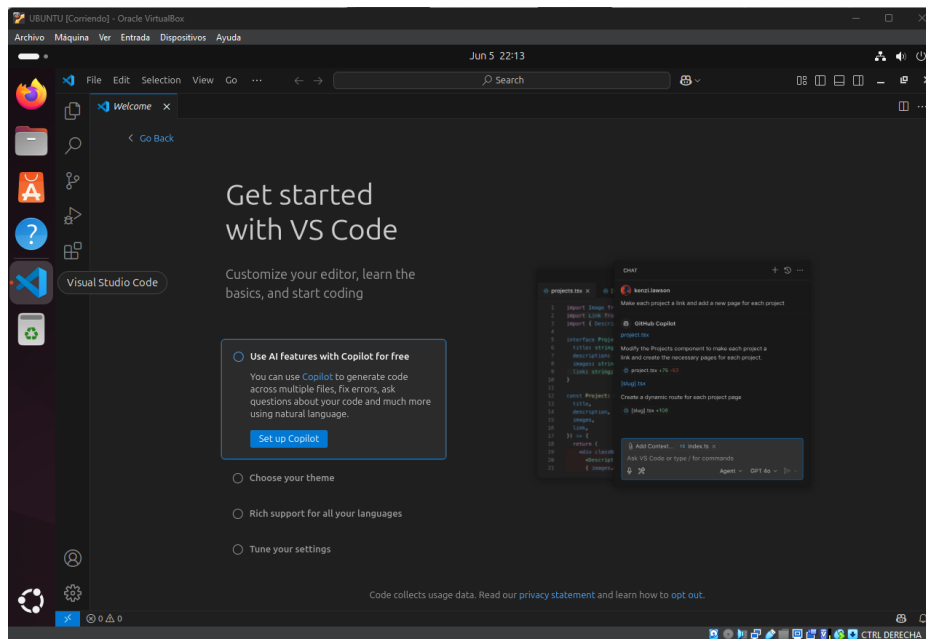
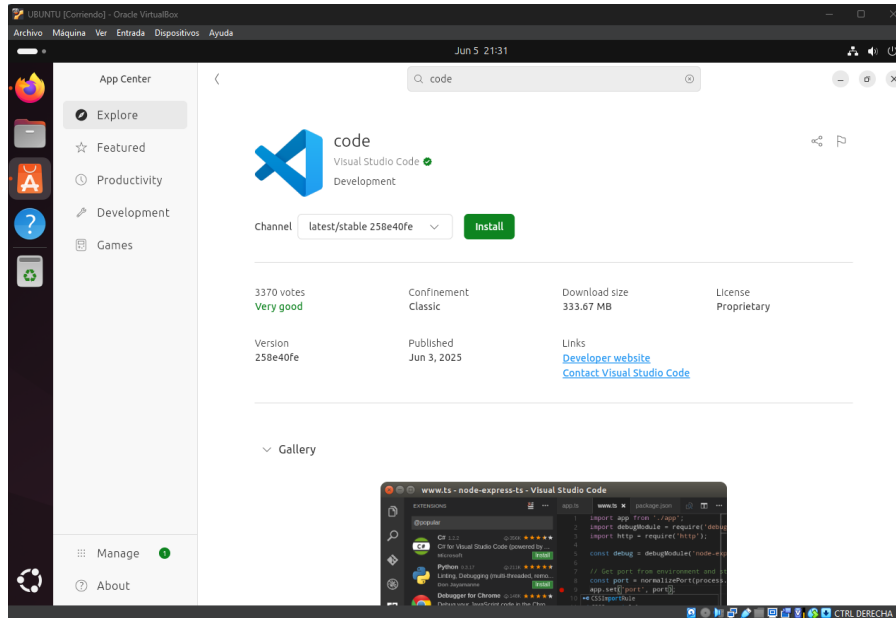
## Quinto paso: Inicio de ubuntu



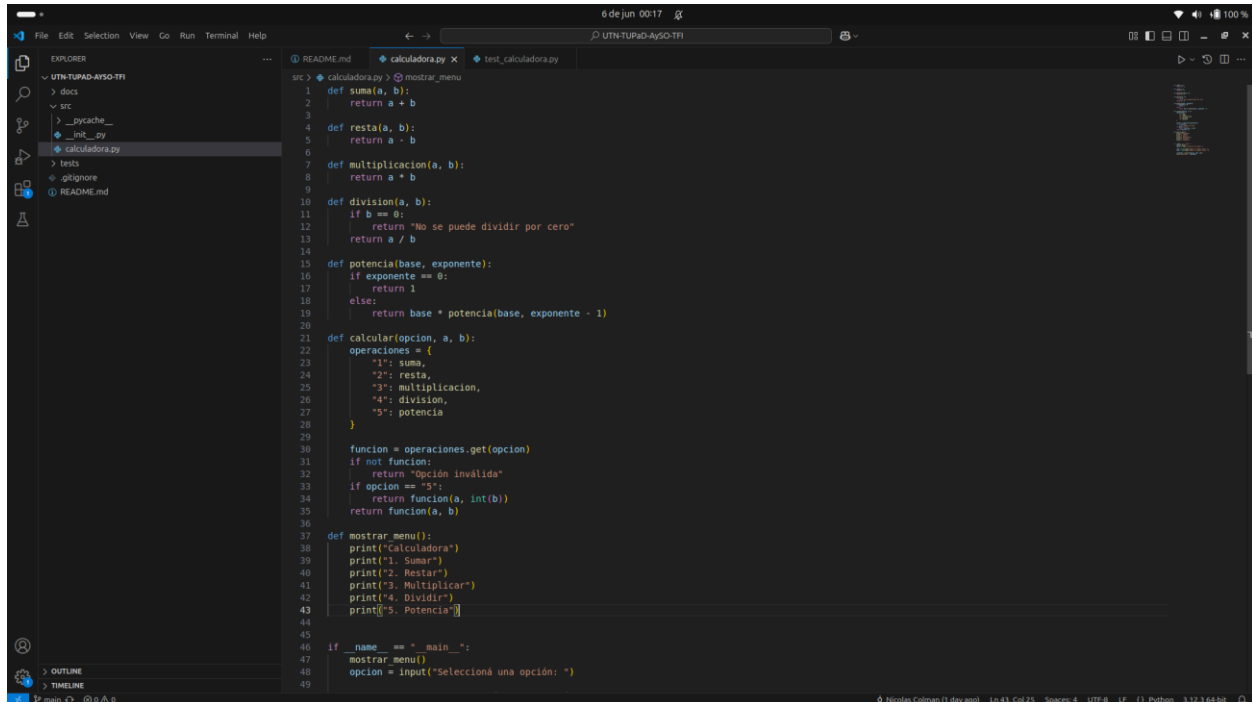
## Sexto paso: Instalación de Visual Studio Code en Ubuntu



# TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

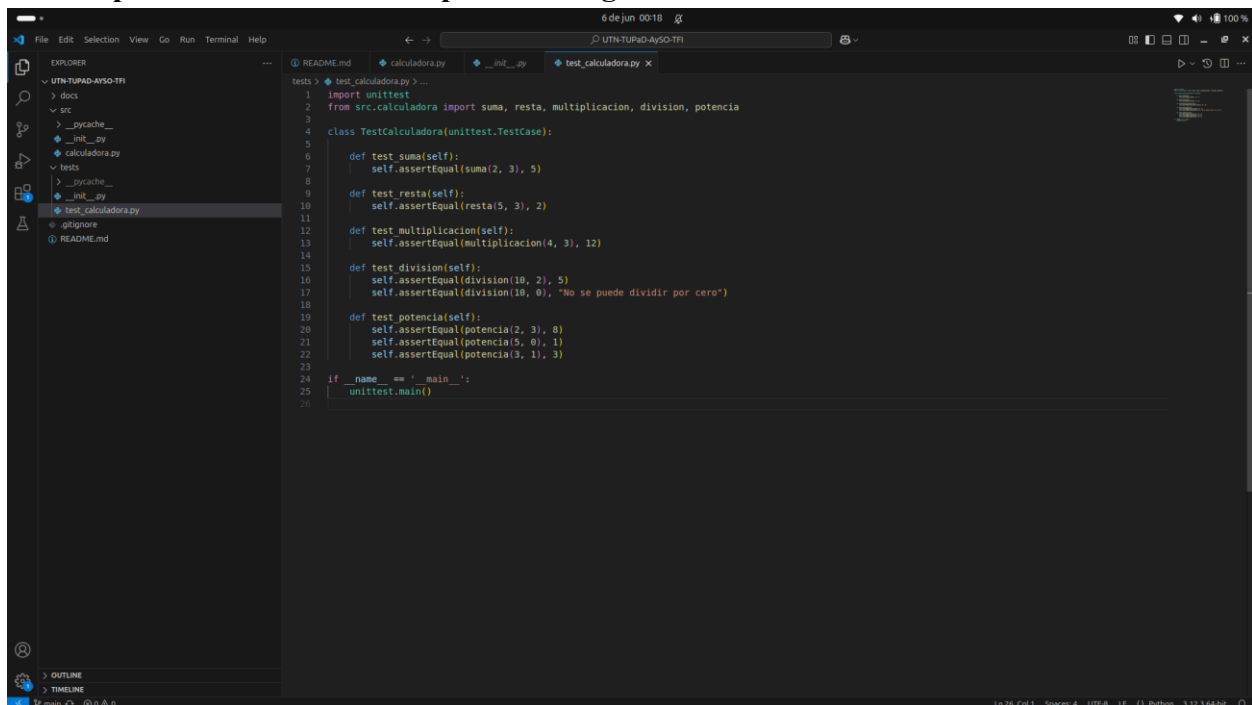


## Séptimo paso: Desarrollo del código



```
1 def suma(a, b):
2     return a + b
3
4 def resta(a, b):
5     return a - b
6
7 def multiplicacion(a, b):
8     return a * b
9
10 def division(a, b):
11     if b == 0:
12         return "No se puede dividir por cero"
13     return a / b
14
15 def potencia(base, exponente):
16     if exponente == 0:
17         return 1
18     else:
19         return base * potencia(base, exponente - 1)
20
21 def calcular(opcion, a, b):
22     operaciones = {
23         "1": suma,
24         "2": resta,
25         "3": multiplicacion,
26         "4": division,
27         "5": potencia
28     }
29
30     funcion = operaciones.get(opcion)
31     if not funcion:
32         return "Opción inválida"
33     if opcion == "5":
34         return funcion(a, int(b))
35     return funcion(a, b)
36
37 def mostrar_menu():
38     print("Calculadora")
39     print("1. Sumar")
40     print("2. Restar")
41     print("3. Multiplicar")
42     print("4. Dividir")
43     print("5. Potencia")
44
45
46 if __name__ == "__main__":
47     mostrar_menu()
48     opcion = input("Seleccioná una opción: ")
49
```

## Octavo paso: Desarrollo de test para el código



```
1 import unittest
2 from src.calculadora import suma, resta, multiplicacion, division, potencia
3
4 class TestCalculadora(unittest.TestCase):
5
6     def test_suma(self):
7         self.assertEqual(suma(2, 3), 5)
8
9     def test_resta(self):
10         self.assertEqual(resta(5, 3), 2)
11
12     def test_multiplicacion(self):
13         self.assertEqual(multiplicacion(4, 3), 12)
14
15     def test_division(self):
16         self.assertEqual(division(10, 2), 5)
17         self.assertEqual(division(10, 0), "No se puede dividir por cero")
18
19     def test_potencia(self):
20         self.assertEqual(potencia(2, 3), 8)
21         self.assertEqual(potencia(5, 0), 1)
22         self.assertEqual(potencia(3, 1), 3)
23
24 if __name__ == "__main__":
25     unittest.main()
26
```

## 3. Metodología Utilizada

- Descarga de VirtualBox oficial de Oracle.



- Instalación de VirtualBox en Windows.
- Descarga de imagen ISO oficial de Ubuntu.
- Configuración de la máquina virtual con Ubuntu como Sistema Operativo.
- Instalación de Ubuntu.
- Instalación de Visual Studio Code.
- Desarrollo del programa de calculadora en Python.
- Desarrollo de los test para las funciones matemáticas.

#### **4. Resultados Obtenidos**

- El sistema operativo Ubuntu se instaló correctamente dentro de la máquina virtual de VirtualBox.
- Verificamos que Python estaba instalado de forma predeterminada.
- Se instaló VSCode para poder desarrollar el código de la calculadora.
- Los tests ejecutan bien el código realizado.

#### **5. Conclusiones**

##### **Aprendizaje**

A lo largo de este trabajo se adquirieron conocimientos fundamentales sobre virtualización, entornos aislados y el uso de hipervisores tipo 2. Se comprendió cómo instalar y configurar máquinas virtuales utilizando VirtualBox y cómo interactúan el sistema operativo anfitrión y el sistema operativo invitado. También se aprendió a instalar programas dentro de la máquina virtual y a utilizarla para poder desarrollar código.

##### **Mejoras**

Si bien se logró el objetivo principal, se identificaron oportunidades de mejora para futuros proyectos. Por ejemplo, se podría automatizar parte del proceso de instalación, probar otras distribuciones de Linux como CentOS o Debian. También sería útil profundizar en aspectos más avanzados como redes virtuales internas, instalación de paquetes adicionales o más integraciones con herramientas de desarrollo.

## **Dificultades**

Durante el desarrollo del proyecto surgieron algunos obstáculos relacionados con la instalación de VirtualBox, y ciertos detalles técnicos en la instalación de servicios dentro de Ubuntu. Estas dificultades sirvieron como desafíos que permitieron afianzar los conocimientos a través de la resolución de problemas concretos.

## **Beneficios**

La virtualización demostró ser una herramienta muy poderosa para crear entornos de prueba y aprendizaje sin necesidad de contar con múltiples equipos físicos. Permite trabajar de forma segura, aislando los cambios y errores del entorno principal. VirtualBox, además, resultó ser una solución accesible y completa para comenzar a aplicar estos conceptos desde una computadora personal, favoreciendo la práctica autónoma y la exploración de nuevos sistemas.

## **6. Bibliografía**

- Stallings, W. (2012). Sistemas Operativos. Pearson Educación.
- Ubuntu Server Guide. Link: <https://ubuntu.com/server/docs>
- Oracle. (s.f.). VirtualBox. Link: <https://www.virtualbox.org/manual>
- Redes virtuales con VBox. Link: <https://wiki.archlinux.org/title/VirtualBox>
- Canonical. (s.f.). Ubuntu Desktop. Link: <https://ubuntu.com/download/desktop>
- Virtualización de recursos por un hipervisor tipo 2. (s.f.). Documento proporcionado por la cátedra.
- GitHub Docs. (s.f.). Introduction to GitHub. Recuperado de <https://docs.github.com/>
- Microsoft. (s.f.). Visual Studio Code. Link: <https://code.visualstudio.com/>
- YouTube - Virtualizacion Conceptos Basicos y Beneficios. Link: <https://www.youtube.com/watch?v=y4o3ablCeco>
- Youtube - Tutorial TFI - Virtualizacion - Arquitectura y Sistemas Operativos. Link: <https://www.youtube.com/watch?v=Ne-IS7m9HEw>

## **7. Anexos**

- Capturas de pantallas de los procesos aplicados. Link: <https://github.com/ncolman94/UTN-TUPaD-AySO-TFI/tree/main/docs>
- Repositorio GitHub con código. Link: <https://github.com/ncolman94/UTN-TUPaD-AySO-TFI>



- Youtube - Video explicativo TFI: <https://youtu.be/jWBAoMF6GYy>