

## **Trabajo Práctico Integrador**

### **Virtualización**

#### **Alumnos**

Jennifer Franco (jennyfranco31.jf@gmail.com)

Jonathan Franco (nahuelfranco7@icloud.com)

**Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.**

#### **Arquitectura y Sistemas Operativos**

##### **Docente Titular**

Osvaldo Falabella

##### **Docente Tutor**

Patricio Costello

05 de Junio de 2025

# Índice

<b>Índice</b>	<b>1</b>
<b>1. Introducción</b>	<b>2</b>
<b>2. Marco Teórico</b>	<b>2</b>
¿Qué es la virtualización?	2
¿Cuándo surge la virtualización?	2
Ventajas de la virtualización	2
¿Qué desafíos o limitaciones puede presentar la virtualización?	3
¿Qué es un hipervisor?	4
Tipos de hipervisores	4
¿Qué es una máquina virtual?	4
¿Qué es VirtualBox?	5
¿Qué es una Imagen ISO?	5
¿Qué es Ubuntu?	5
Ventajas de Ubuntu	5
¿Qué son los snapshots?	5
Redes virtuales en la virtualización	6
¿Cómo se aplica la virtualización en el trabajo de un técnico en programación?	6
<b>3. Caso Práctico</b>	<b>6</b>
Paso 1. Verificar la compatibilidad de virtualización en el equipo anfitrión	7
Paso 2. Descargar VirtualBox	7
Paso 3. Descargar Ubuntu Server	14
Paso 4. Crear la máquina virtual en VirtualBox	15
Paso 5. Asignar memoria RAM y procesadores	16
Paso 6. Configurar el disco duro virtual	17
Paso 7. Confirmar la configuración de la máquina virtual	18
Paso 8. Iniciar la máquina virtual	19
Paso 9. Descargar Visual Studio Code	20
Paso 10. Verificar si Python se encuentra instalado	22
Paso 11. Descargamos la extensión de Python en Visual Studio Code	22
Paso 12. Creación del archivo calculador_promedio.py	23
Paso 13. Desarrollo del código.	25
<b>4. Metodología Utilizada</b>	<b>27</b>
Investigación previa	27
Herramientas utilizadas	28
Trabajo colaborativo	28
<b>5. Resultados Obtenidos</b>	<b>28</b>
<b>6. Conclusiones</b>	<b>29</b>
<b>7. Bibliografía</b>	<b>29</b>
<b>8. Anexo</b>	<b>30</b>
Código del programa	30
Video explicativo	31
Repositorio GitHub	31

# 1. Introducción

En la actualidad, la virtualización es una tecnología clave ya que permite optimizar recursos físicos al posibilitar la ejecución de múltiples sistemas operativos en una misma máquina. Comprender la virtualización y adquirir habilidades prácticas en el uso de herramientas como Oracle VirtualBox resulta esencial para desenvolverse con eficacia en proyectos de software modernos.

Este trabajo se centra en el estudio de la virtualización como herramienta fundamental en entornos de desarrollo, prueba y despliegue de software. La elección de este tema se debe a la gran relevancia que posee en la programación, donde es habitual trabajar con distintos sistemas operativos, entornos aislados y configuraciones específicas que no siempre pueden replicarse en una máquina física única.

Entre los objetivos principales del trabajo se encuentran conocer los conceptos básicos de la virtualización, aprender a instalar y configurar una máquina virtual, y reflexionar sobre las ventajas que esta tecnología aporta al proceso de desarrollo de software.

## 2. Marco Teórico

### ¿Qué es la virtualización?

El concepto de virtualización puede significar muchos términos, aplicado a los sistemas operativos, consiste básicamente en poder compartir una misma infraestructura de Hardware por varios sistemas operativos funcionando de forma totalmente independiente.

Teniendo un mismo servidor, con un mismo disco duro, un mismo procesador y una misma memoria, se puede disponer de varias instalaciones de Windows y Linux funcionando en paralelo, totalmente independientes las unas de las otras. Si una se para o tiene problemas, las otras no sufren alteraciones y podrían aprovechar los recursos de velocidad de proceso que quedarían liberados (Muñoz, s.f.).

### ¿Cuándo surge la virtualización?

En la década de 1960, la virtualización surge como respuesta a la necesidad de optimizar el alto costo de los recursos de hardware, principalmente en entornos de computación de grandes empresas y universidades. Fue desarrollada inicialmente por IBM, que buscaba formas de dividir la potencia de procesamiento de sus mainframes (grandes computadoras centrales) entre múltiples usuarios.

### Ventajas de la virtualización

1. Ahorro de costes: Es la razón principal por la que las empresas implementan la virtualización, supone un ahorro de dinero al necesitar comprar menos hardware porque una máquina puede realizar el trabajo de varias.
2. Compatibilidad de programas: La virtualización de sistemas operativos, como Windows sobre entornos Linux o macOS, permite ejecutar aplicaciones nativas no

disponibles para estas plataformas. Esta solución garantiza la interoperabilidad de software y evita recurrir a desarrollos alternativos o mantener configuraciones duales.

3. Entornos de prueba: La virtualización permite crear entornos seguros para la evaluación de software en versiones beta o de programas que se desean probar antes de su implementación definitiva. Asimismo, es posible instalar y explorar nuevos sistemas operativos, como versiones recientes de Windows, sin comprometer la integridad del sistema operativo principal instalado en el equipo físico ya sea Windows, Linux o macOS.
4. Aislamiento y seguridad: Las máquinas virtuales son totalmente independientes entre sí, por tanto un fallo en una aplicación o en una máquina virtual afectará únicamente a esa máquina virtual, y el resto de máquinas virtuales seguirán funcionando normalmente. De la misma forma cada máquina tiene un acceso privilegiado (root o administrador) independiente, por lo que un ataque de seguridad en una máquina virtual sólo afectará a esa máquina.
5. Flexibilidad y agilidad: La virtualización permite configurar máquinas virtuales con recursos personalizados, como CPU, memoria, almacenamiento y red, sin necesidad de adquirir hardware físico adicional. Además, el proceso de creación de una máquina virtual es sumamente ágil, pudiendo completarse mediante comandos simples. Esto permite desplegar nuevos entornos, como servidores, casi de forma inmediata y sin los tiempos asociados a la compra y configuración de equipos físicos.

¿Qué desafíos o limitaciones puede presentar la virtualización?

1. Rendimiento inferior: La ejecución simultánea de varios sistemas operativos virtualizados no alcanza el nivel de rendimiento que se obtendría al ejecutar dichos sistemas directamente sobre el hardware físico. Dado que el virtualizador introduce una capa intermedia en la gestión del hardware para gestionar las peticiones de acceso y la concurrencia al mismo, el rendimiento de la máquina virtual se ve afectado irremediablemente.
2. Limitaciones en el Hardware: No es posible utilizar hardware que no esté gestionado o soportado por el hipervisor.
3. Proliferación de máquinas virtuales: La facilidad para crear máquinas virtuales sin necesidad de adquirir hardware adicional puede llevar a una proliferación descontrolada de instancias. Esto incrementa considerablemente las tareas de administración, los costos de licencias y los riesgos de seguridad asociados a una infraestructura compleja y poco controlada.
4. Desaprovechamiento de recursos: Crear máquinas virtuales innecesarias tiene un coste en ocupación de recursos, principalmente en espacio en disco, RAM y capacidad de proceso.

5. Centralización de las máquinas en un único servidor: La centralización de múltiples máquinas virtuales en un único servidor físico genera un punto único de fallo, si existe una avería en el servidor anfitrión compromete todas las instancias virtuales que aloja. Para mitigar este riesgo, es necesario implementar soluciones de alta disponibilidad, como clústeres o mecanismos de replicación, lo que implica una inversión económica adicional.
6. Portabilidad limitada entre virtualizadores: Como cada producto de virtualización usa su propio sistema, no hay uniformidad o estandarización de formatos y la portabilidad entre plataformas está condicionada a la solución de virtualización adoptada. Elegir GNU/Linux, Mac OS X, Windows o Solaris como anfitrión es una decisión importante en entornos corporativos.

### ¿Qué es un hipervisor?

Un hipervisor, también llamado monitor de máquina virtual, es un software que permite crear y gestionar máquinas virtuales, actúa como intermediario entre el hardware físico y los sistemas operativos que se ejecutan en las máquinas virtuales, asegurando que cada una de ellas tenga acceso controlado a los recursos.

### Tipos de hipervisores

Los hipervisores se pueden clasificar en dos tipos:

1. Tipo 1 (nativo, bare-metal): El software se ejecuta directamente sobre el hardware real del equipo para controlar los recursos y supervisar la ejecución de los sistemas operativos virtualizados. Estos últimos operan en un nivel superior al del hipervisor, que actúa como intermediario entre el hardware y las máquinas virtuales.

Algunos de los hipervisores de tipo 1 más conocidos son los siguientes:

- VMware: ESXi, ESX.
- Xen.
- Citrix XenServer.
- Microsoft Hyper-V Server.

2. Tipo 2 (Hosted): Es una aplicación que se ejecuta sobre un sistema operativo convencional (Linux, Windows, MacOS) y permite virtualizar otros sistemas operativos. De esta forma la virtualización se produce en una capa más alejada del hardware, comparado con los hipervisores de tipo 1, implica una reducción en el rendimiento debido a la sobrecarga del sistema base.

Algunos de los hipervisores de tipo 2 más utilizados son los siguientes:

- Sun: VirtualBox, VirtualBox OSE.
- VMware: Workstation, Server, Player.
- Microsoft: Virtual PC, Virtual Server

### ¿Qué es una máquina virtual?

Es un entorno virtual que simula una computadora física, permitiendo instalar y ejecutar sistemas operativos como si fueran independientes, pero usando los recursos del equipo real.

## ¿Qué es VirtualBox?

El software de virtualización multiplataforma de código abierto más popular del mundo, permite a los desarrolladores entregar código más rápido, ya que pueden ejecutar múltiples sistemas operativos en un solo dispositivo. Los equipos de TI y los proveedores de soluciones usan VirtualBox para reducir los costos operativos y acortar el tiempo necesario para implementar aplicaciones de forma segura en entornos locales y en la nube (Oracle, 2023).

## ¿Qué es una Imagen ISO?

Una imagen ISO es un archivo único que representa el contenido completo de un CD o DVD, el cual contiene todo el contenido del disco en un solo archivo, incluyendo el sistema de archivos. Se utilizan comúnmente para distribuir sistemas operativos y paquetes de software, y son compatibles con software de virtualización para la instalación de sistemas operativos invitados (Oracle, 2023).

## ¿Qué es Ubuntu?

Según Boada (2025), Ubuntu es un popular sistema operativo gratuito y de código abierto basado en Linux, lo que significa que los usuarios pueden modificar su código, crear e instalar tantas copias como deseen y distribuir la personalización en cualquier lugar. Ubuntu fue presentado en 2004 por la empresa británica Canonical. Se basaba en Debian, una distro muy popular por aquel entonces, que era difícil de instalar. Por ello, Ubuntu se propuso como una alternativa más fácil de usar.

## Ventajas de Ubuntu

1. Facilidad de uso: Ubuntu tiene una interfaz sencilla e intuitiva.
2. Fuerte seguridad: Ubuntu emplea AppArmor y medidas de seguridad avanzadas para evitar infracciones.
3. Más opciones de software: Ubuntu tiene un enorme número de aplicaciones que puedes instalar, muchas de las cuales están disponibles exclusivamente para el sistema operativo.
4. Mayor privacidad: Ubuntu garantiza una estricta política de privacidad de datos y permite a los usuarios personalizar la configuración de privacidad.
5. Rendimiento ligero: la interfaz predeterminada de Ubuntu sólo utiliza menos de 1 GB de RAM. Por lo tanto, el sistema operativo funciona en dispositivos de gama baja.
6. Gratuidad: Ubuntu es una distribución de Linux de código abierto y gratuito (Boada, 2025).

## ¿Qué son los snapshots?

Los snapshots son una característica de las máquinas virtuales que permiten guardar el estado completo de una máquina virtual en un momento determinado, esto incluye el estado del sistema operativo, las aplicaciones abiertas, la configuración de hardware virtual y los datos almacenados en ese instante. Resulta sumamente útil ya que permite restaurar rápidamente el entorno a un punto anterior en caso de fallos o errores, sin necesidad de realizar una nueva instalación o configuración, lo que ahorra tiempo y recursos. Además, se

pueden guardar múltiples snapshots, creando una especie de historial de estados para un seguimiento detallado de los cambios realizados.

### Redes virtuales en la virtualización

Las redes virtuales permiten la comunicación entre las máquinas virtuales, con el host físico y con redes externas, configurando distintas formas de conexión según las necesidades del entorno virtualizado.

Los principales tipos de redes virtuales que existen son:

1. NAT (Network Address Translation): En este modo, la máquina virtual comparte la conexión de red del host a través de la traducción de direcciones IP. La máquina virtual puede acceder a redes externas utilizando la IP del host, pero desde el exterior no es posible acceder directamente a ella.
2. Bridge (Puente): La máquina virtual se conecta directamente a la red física a la que está conectado el host, como si fuera otro dispositivo más en la misma red. La máquina virtual obtiene su propia dirección IP de la red local, y es accesible desde otros dispositivos en la red física, igual que cualquier otro equipo.
3. Host-only: La máquina virtual se conecta a una red virtual aislada que solo existe entre la máquina virtual y el host físico. No hay acceso a redes externas ni a internet.

### ¿Cómo se aplica la virtualización en el trabajo de un técnico en programación?

Como se mencionó anteriormente, la implementación de la virtualización permite crear entornos virtuales que simulan sistemas operativos completos, sin necesidad de utilizar múltiples equipos físicos. Esta capacidad resulta sumamente útil en contextos donde se requiere probar aplicaciones en diferentes plataformas, ya que posibilita la instalación y ejecución de sistemas operativos diversos, como Windows, Linux o macOS, sobre una misma máquina física, garantizando así la compatibilidad y el correcto funcionamiento del software en distintos entornos.

Además, la virtualización facilita la creación de entornos de desarrollo aislados, lo que permite al técnico trabajar en múltiples proyectos sin que se generen conflictos entre dependencias, bibliotecas o versiones de herramientas. Esta práctica contribuye a mantener un entorno de trabajo limpio y ordenado, favoreciendo la estabilidad del sistema operativo principal y la eficiencia en el desarrollo de soluciones informáticas.

## 3. Caso Práctico

Se necesitaba crear un entorno de desarrollo controlado para trabajar con Visual Studio Code y Python de manera segura y aislada del sistema operativo principal. Para ello, se utilizó VirtualBox para instalar una máquina virtual con Ubuntu como sistema operativo invitado.

## Paso 1. Verificar la compatibilidad de virtualización en el equipo anfitrión

Antes de comenzar con la instalación de la máquina virtual, es necesario asegurarse de que el procesador del equipo anfitrión admite virtualización y que dicha función está habilitada en la BIOS o UEFI.

Para ello, en Windows, se puede usar el Administrador de tareas, se abre la pestaña Rendimiento, y se comprueba si “Virtualización” está habilitada, como se muestra en la captura de pantalla. En caso de que se encuentre deshabilitada, se debe reiniciar el equipo para acceder a la BIOS/UEFI (suele ser con las teclas Supr, F2, Esc u otra), se busca en las opciones de configuración avanzada la opción de Intel VT-x, Intel Virtualization Technology o AMD-V, y asegurarse de que esté habilitada, se debe guardar los cambios y salir.

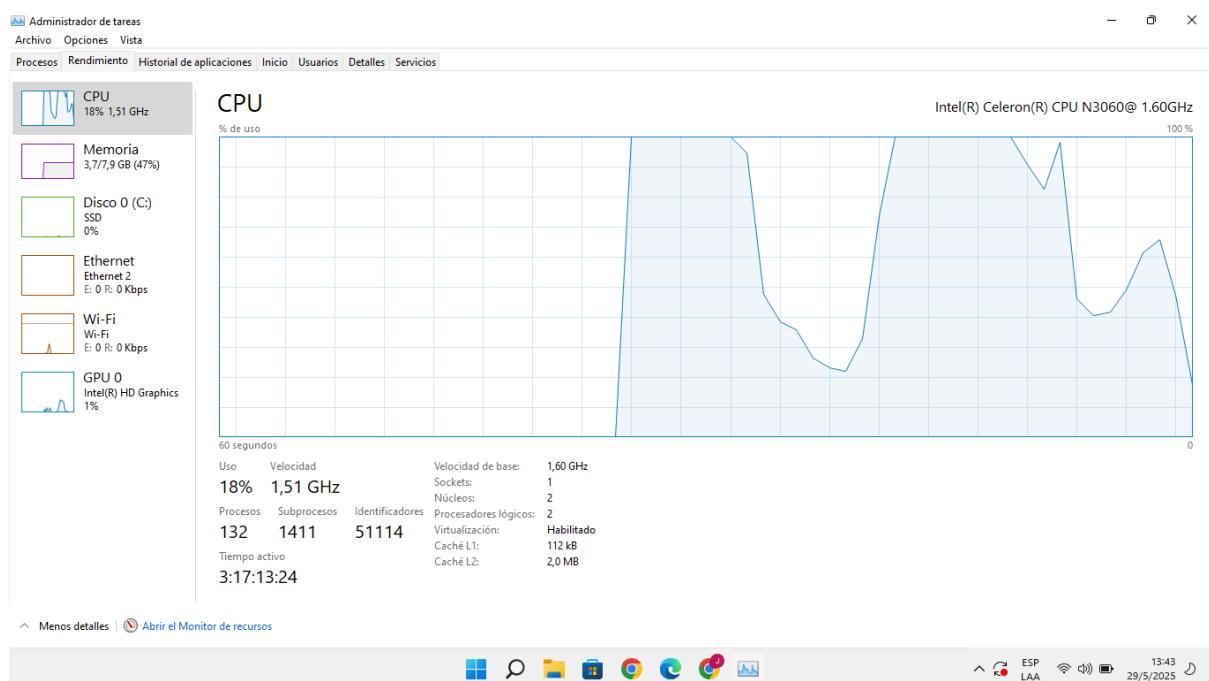


Figura 1. Captura de pantalla del administrador de windows.

## Paso 2. Descargar VirtualBox

1. Se debe acceder a la página oficial de VirtualBox:  
<https://www.virtualbox.org/>
2. Se descarga el instalador para el sistema operativo anfitrión (Windows hosts)
3. Se descarga el paquete de extensión para ampliar las capacidades de VirtualBox, ofreciendo características adicionales que no vienen incluidas en la instalación base.

A continuación se agregan capturas de pantallas del proceso completo de instalación.

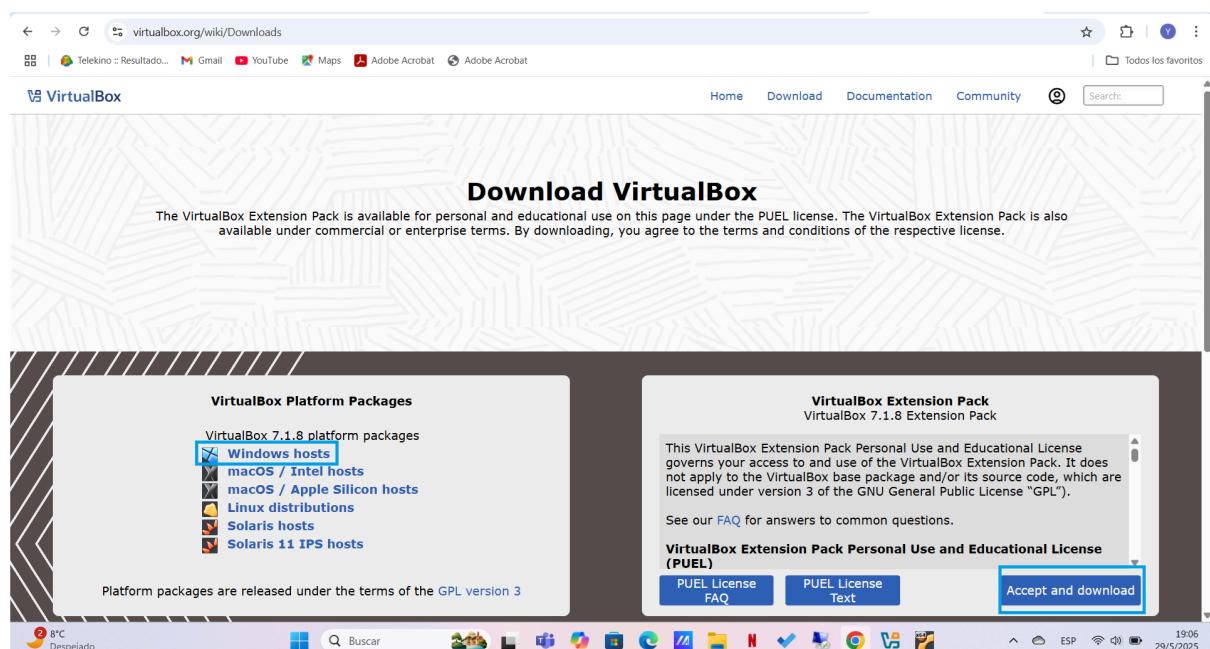


Figura 2. Captura de pantalla de la página oficial de VirtualBox

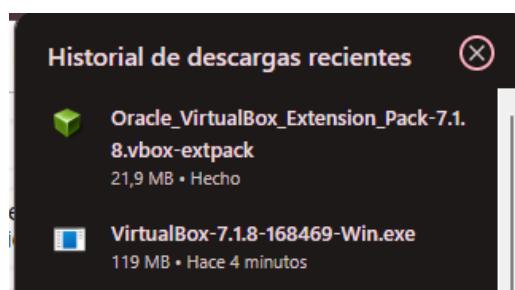


Figura 3. Captura de pantalla del historial de descarga.

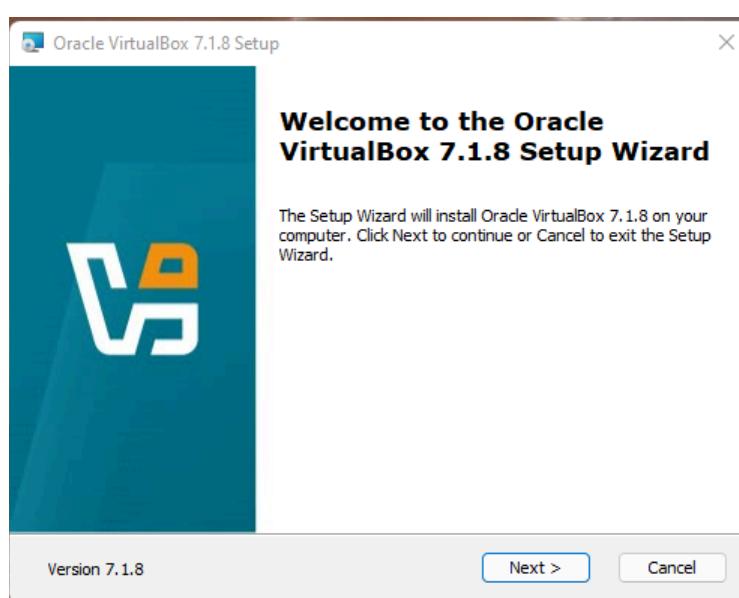


Figura 4. Captura de pantalla del proceso de instalación de VirtualBox.

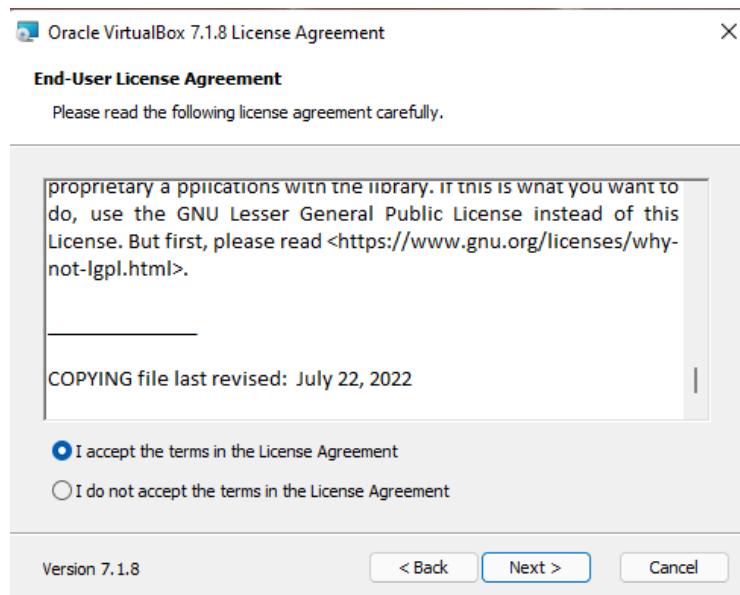


Figura 5. Captura de pantalla del proceso de instalación de VirtualBox, aceptación de términos.

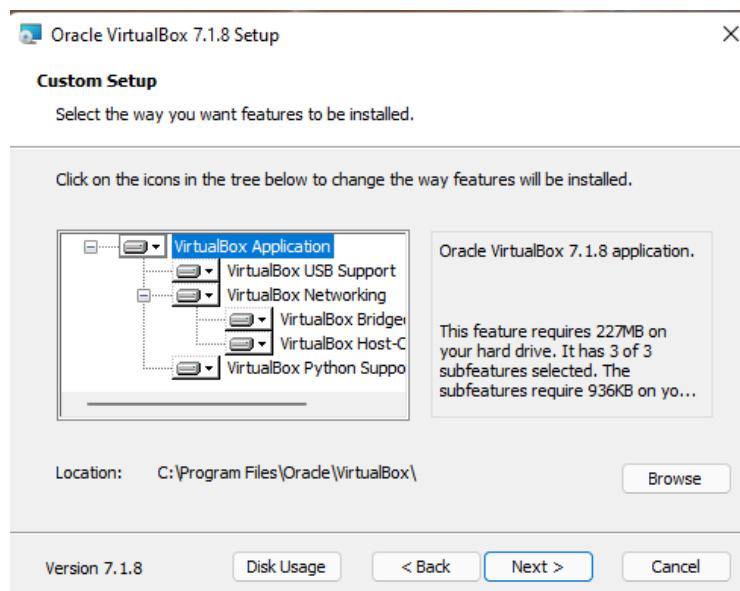


Figura 6. Captura de pantalla del proceso de instalación de VirtualBox, seleccionar la ubicación de guardado.

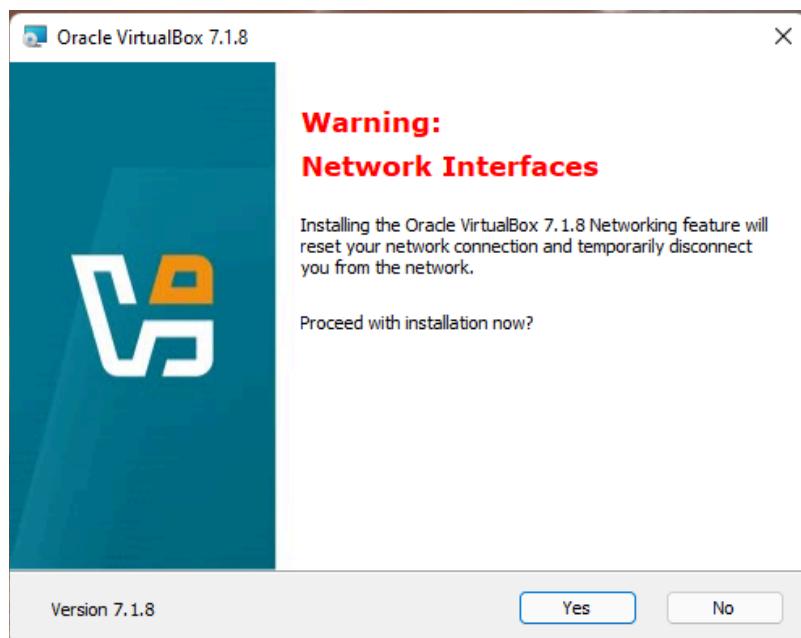


Figura 7. Captura de pantalla del proceso de instalación de VirtualBox, advertencia de instalación de interfaces de red.

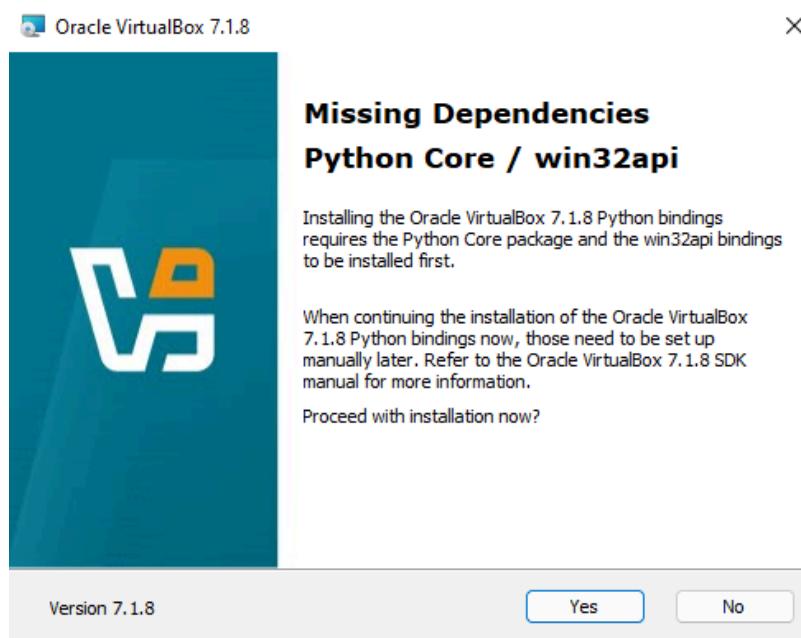


Figura 8. Captura de pantalla del proceso de instalación de VirtualBox, advertencia de dependencias faltantes.

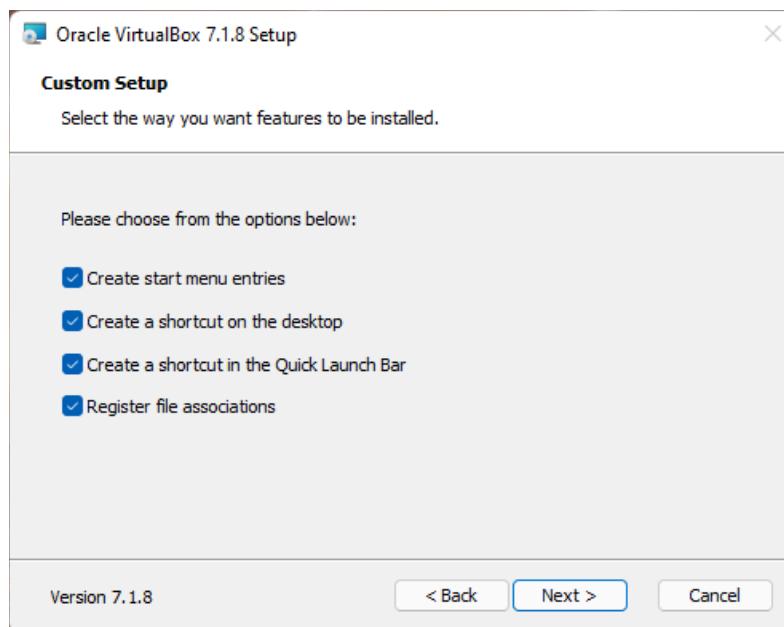


Figura 9. Captura de pantalla del proceso de instalación de VirtualBox, configuración personalizada.

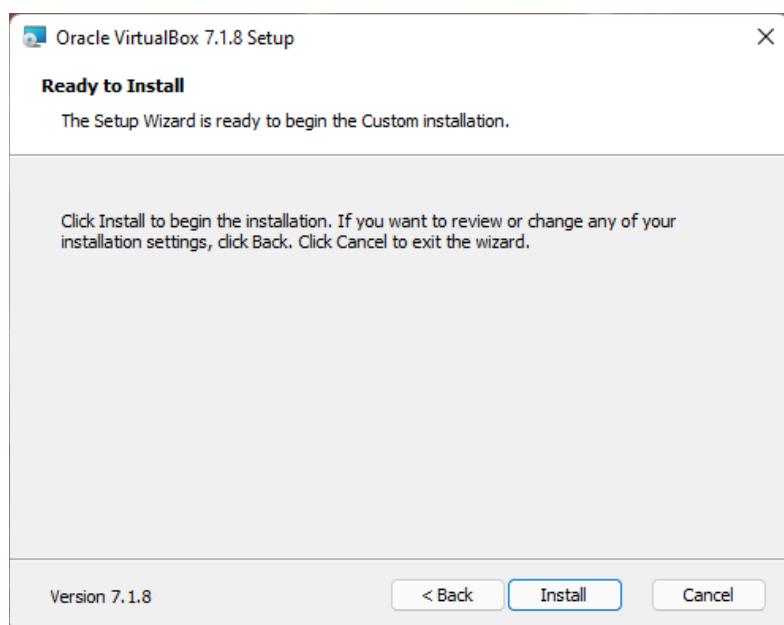


Figura 10. Captura de pantalla del proceso de instalación de VirtualBox, la instalación está lista para comenzar.

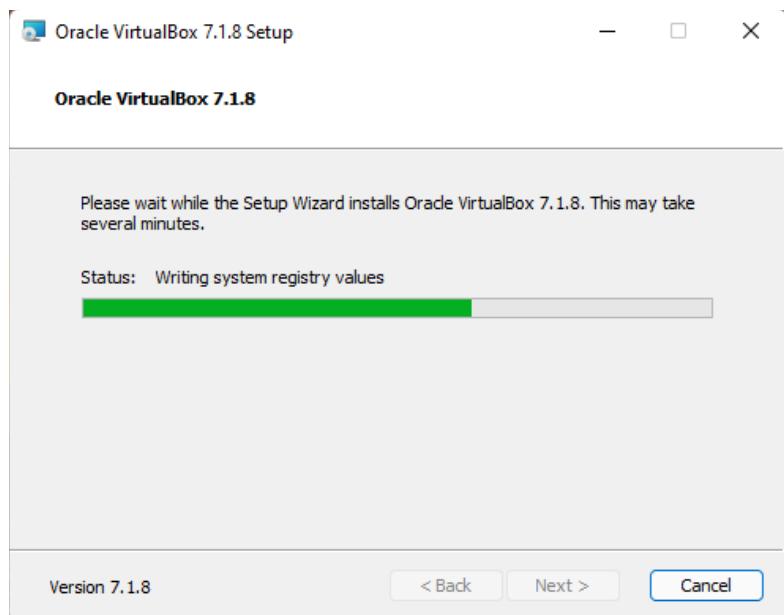


Figura 11. Captura de pantalla del proceso de instalación de VirtualBox.



Figura 12. Captura de pantalla del final del proceso de instalación de VirtualBox.

Las siguientes capturas de pantalla corresponden a la instalación del paquete de extensión de VirtualBox.

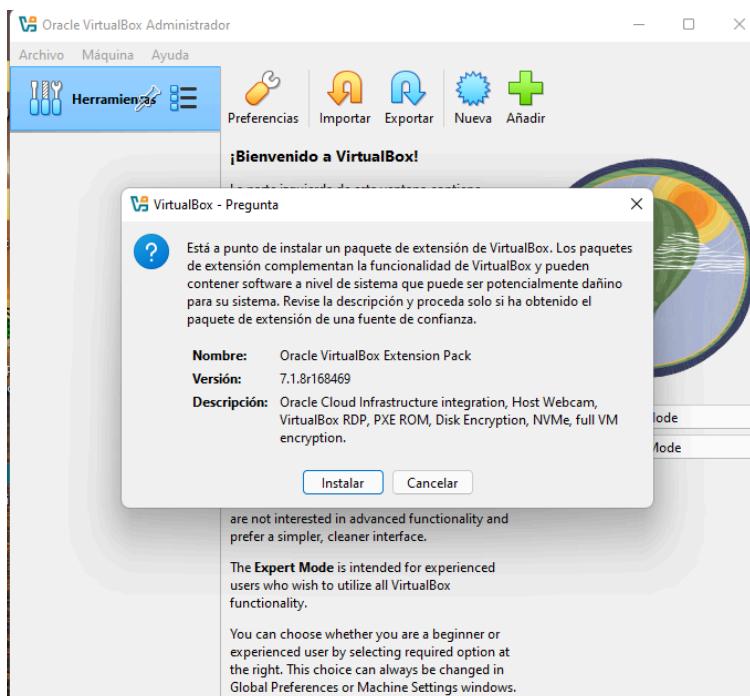


Figura 13. Captura de pantalla del proceso de instalación del Oracle VirtualBox Extension Pack

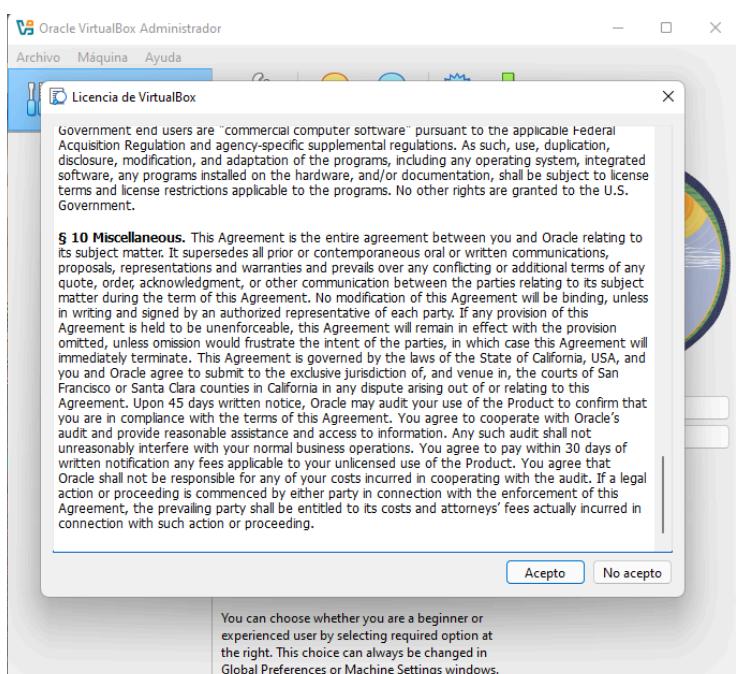


Figura 14. Captura de pantalla del proceso de instalación del Oracle VirtualBox Extension Pack, aceptación de la licencia.

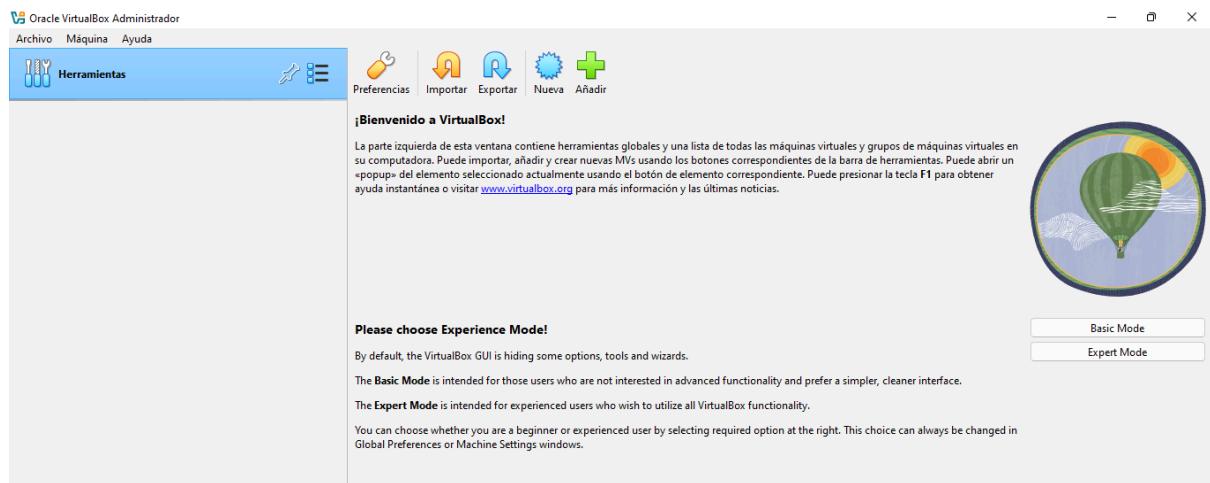


Figura 15. Captura de pantalla del proceso de instalación del Oracle VirtualBox Extension Pack finalizado.

### Paso 3. Descargar Ubuntu Server

1. Acceder a la página oficial de Ubuntu:  
<https://ubuntu.com/download/server>
2. Descargar la imagen ISO de la última versión, en este caso Ubuntu Server 22.04.2 LTS).

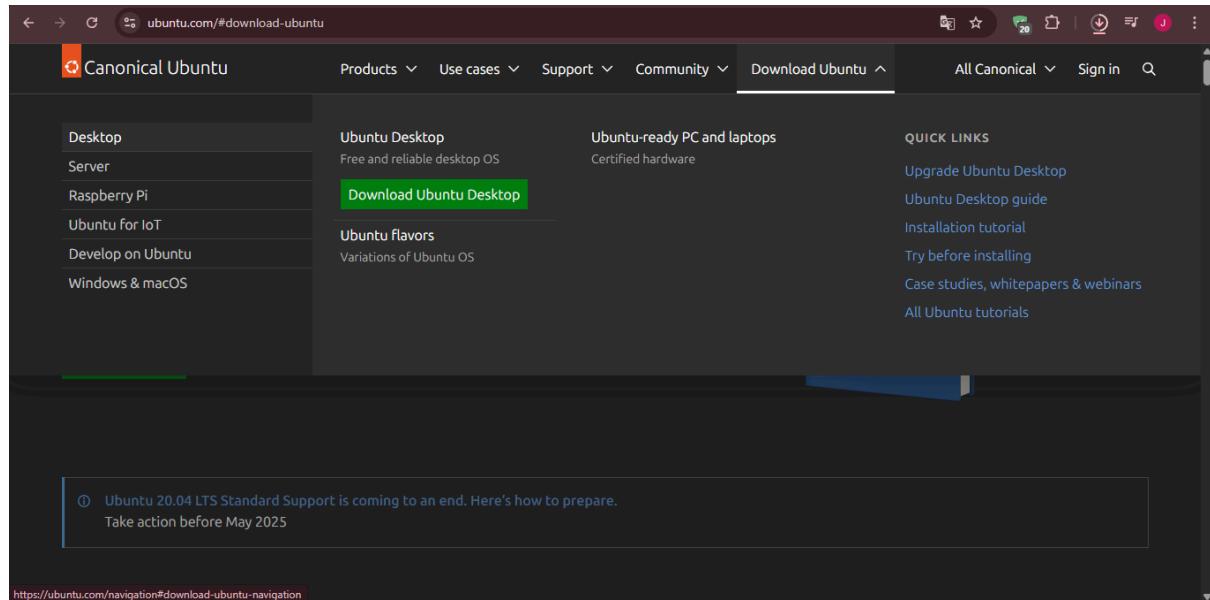


Figura 16. Captura de pantalla del sitio web oficial de Ubuntu.

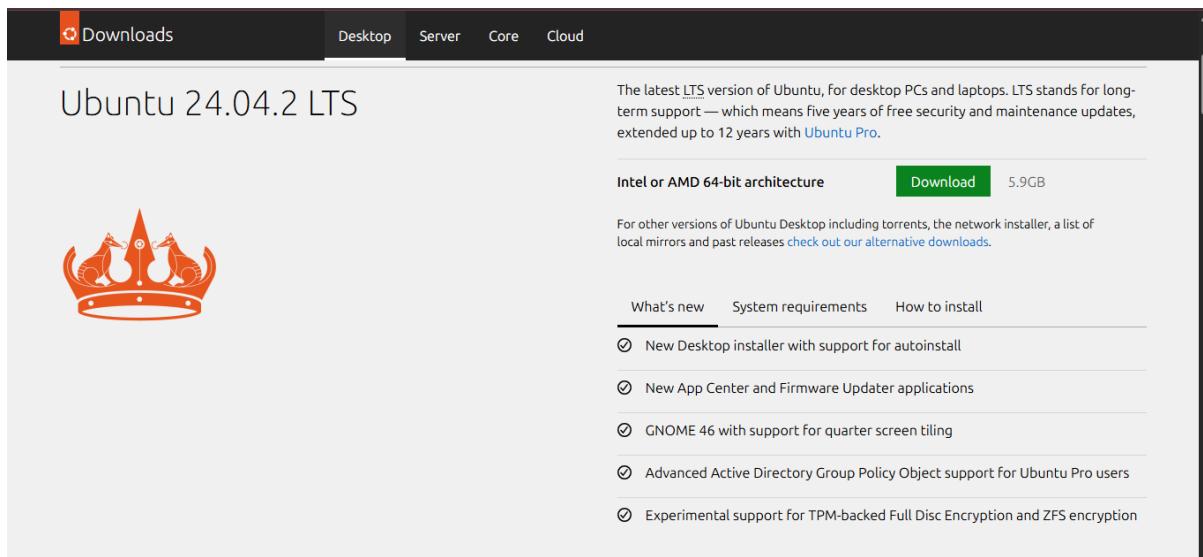


Figura 17. Captura de pantalla del sitio web oficial de Ubuntu en la pestaña desktop.

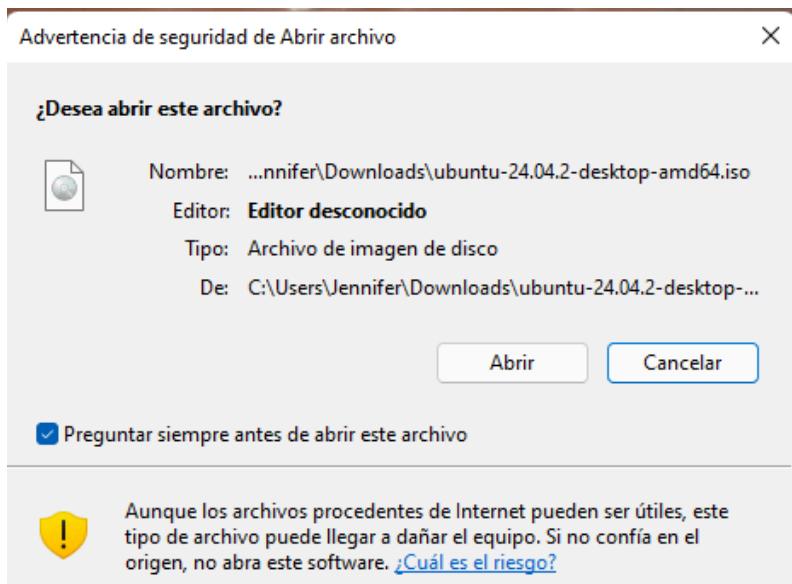


Figura 18. Captura de pantalla de la advertencia de seguridad al intentar abrir el archivo .iso de Ubuntu.

#### Paso 4. Crear la máquina virtual en VirtualBox

1. Abrir VirtualBox.
2. Hacer clic en Nueva.
3. Asignar un nombre descriptivo, en este caso UBUNTU-AYSO.

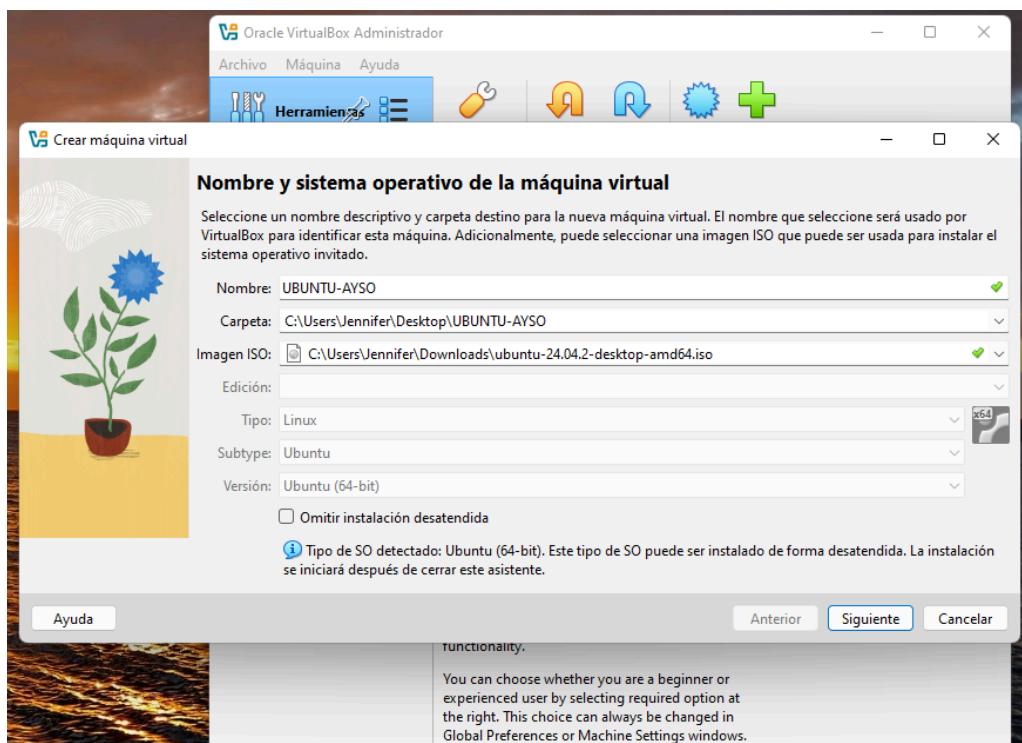


Figura 19. Captura de pantalla de la configuración de la máquina virtual, elección del nombre y sistema operativo.

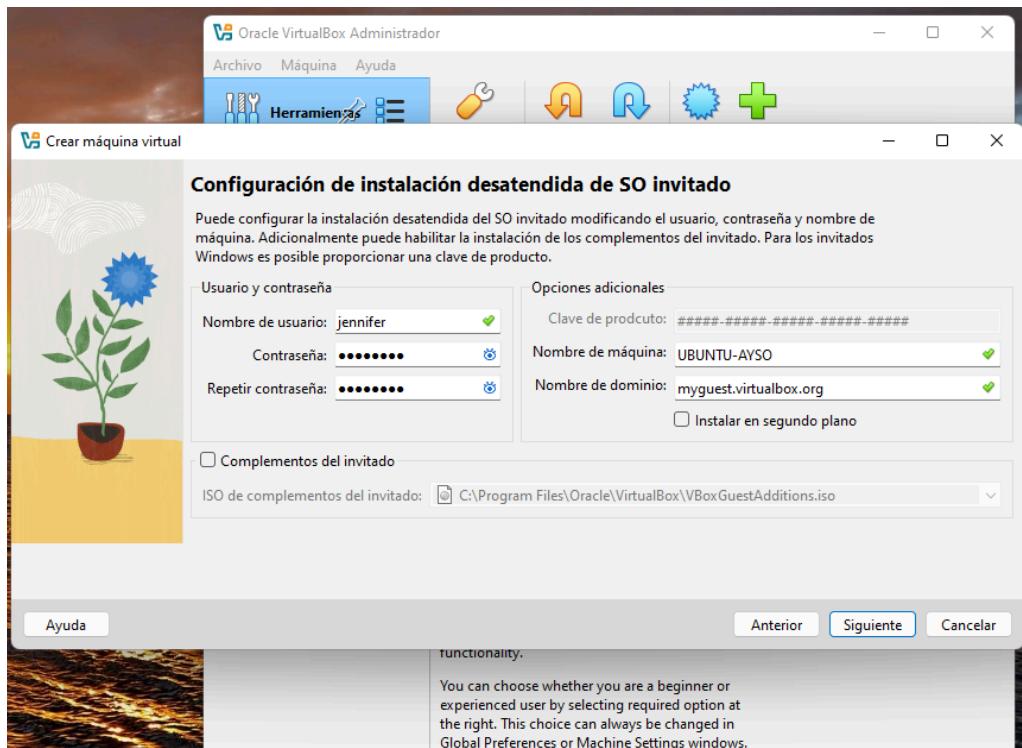


Figura 20. Captura de pantalla de la configuración de la máquina virtual, creación de usuario y contraseña.

## Paso 5. Asignar memoria RAM y procesadores

1. Elegir la cantidad de memoria RAM a asignar, en este caso 3072MB.
2. Asignar la cantidad de núcleos deseada, en este caso 2 CPUs .

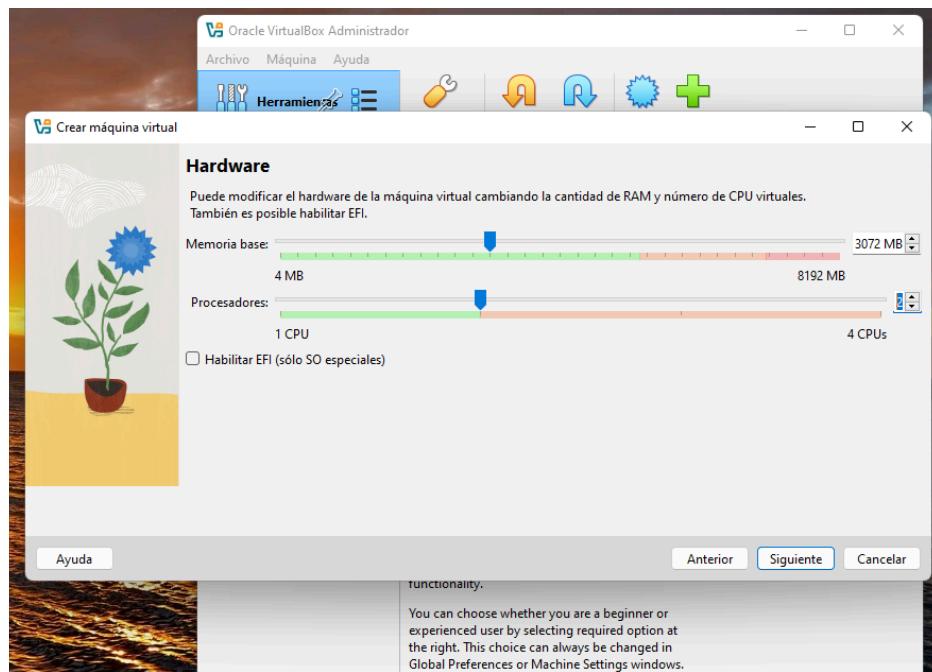


Figura 21. Captura de pantalla de la configuración de la máquina virtual, asignación de hardware.

#### Paso 6. Configurar el disco duro virtual

1. Crear un disco duro virtual.
2. Asignar un tamaño adecuado, en este caso 25GB.

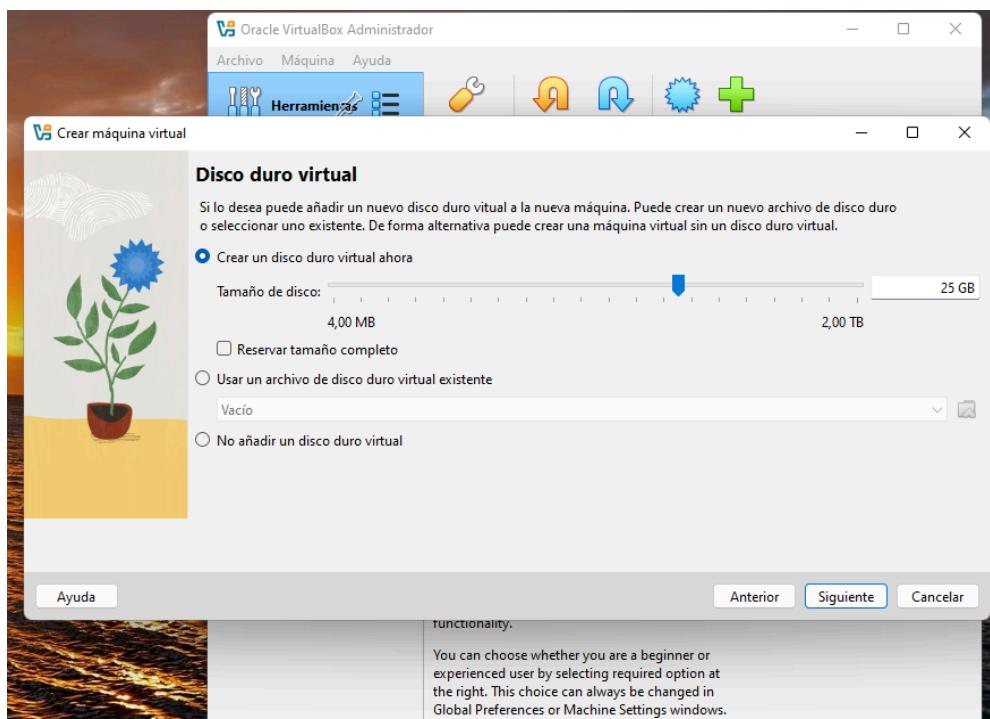


Figura 22. Captura de pantalla de la configuración de la máquina virtual, creación del disco duro virtual.

### Paso 7. Confirmar la configuración de la máquina virtual

En este paso se verifica la memoria RAM, procesadores, disco duro, nombre de la máquina virtual y otros ajustes generales. También se confirma que la imagen ISO de Ubuntu esté asignada como medio de arranque.

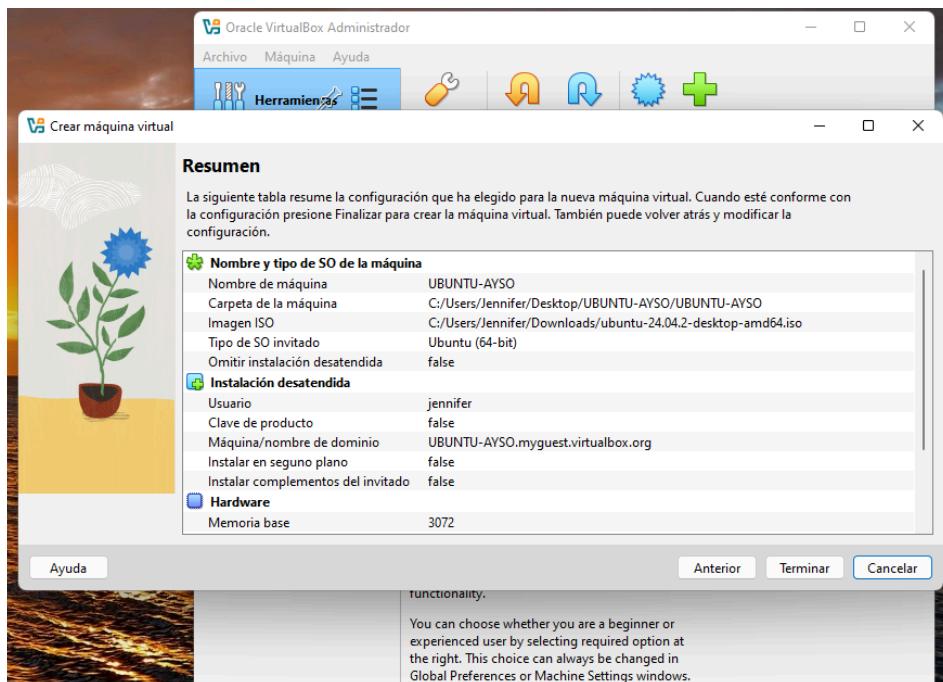


Figura 23. Captura de pantalla de la configuración de la máquina virtual, resumen de la configuración.

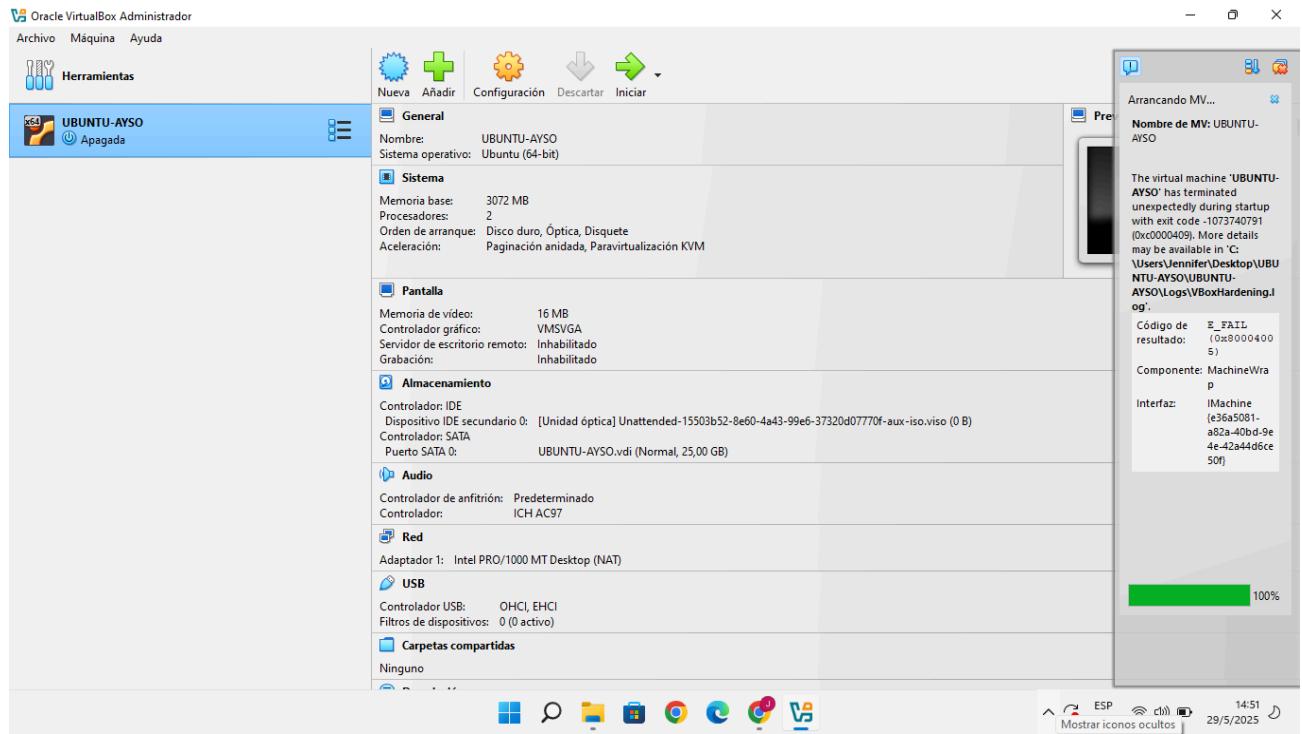


Figura 24. Captura de pantalla de la vista general de la máquina virtual ya creada.

### Paso 8. Iniciar la máquina virtual

Se debe hacer clic en “Iniciar” para arrancar la máquina virtual y comenzar la instalación de Ubuntu Server.

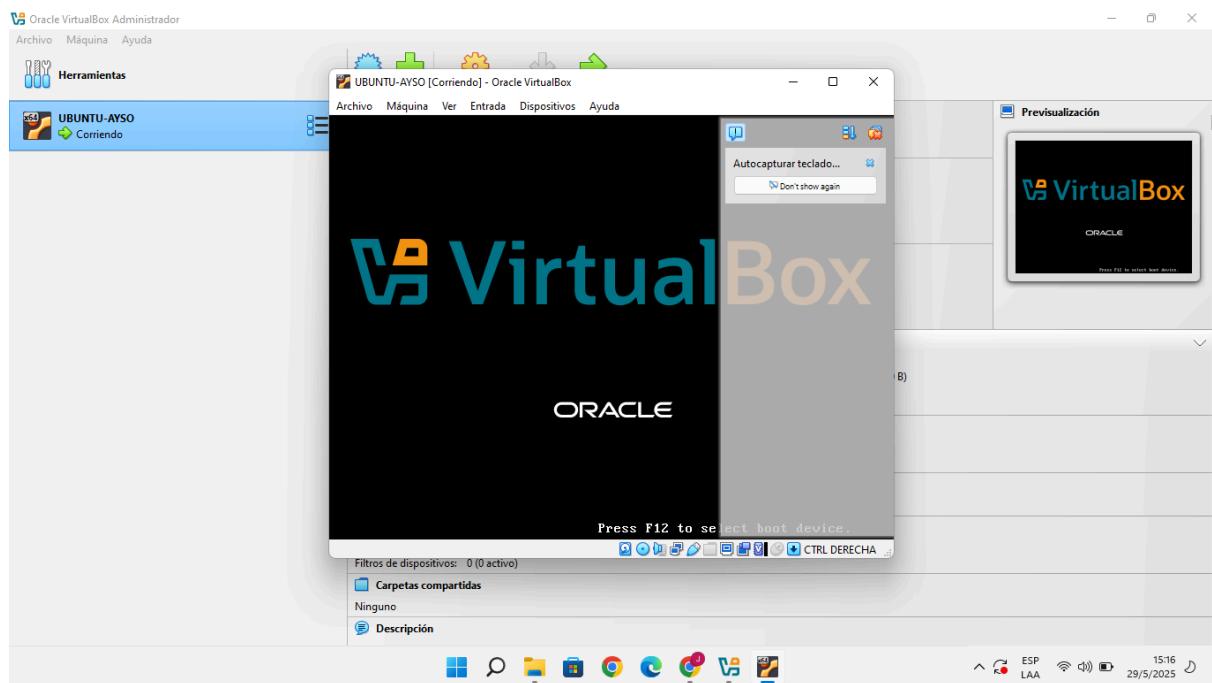


Figura 25. Captura de pantalla del arranque de la máquina virtual UBUNTU-AYSO.

## Paso 9. Descargar Visual Studio Code

En la máquina virtual se debe ingresar a la tienda App Center de Linux para descargar el programa Visual Studio Code, para poder realizar la descarga se le solicitará la clave creada anteriormente.

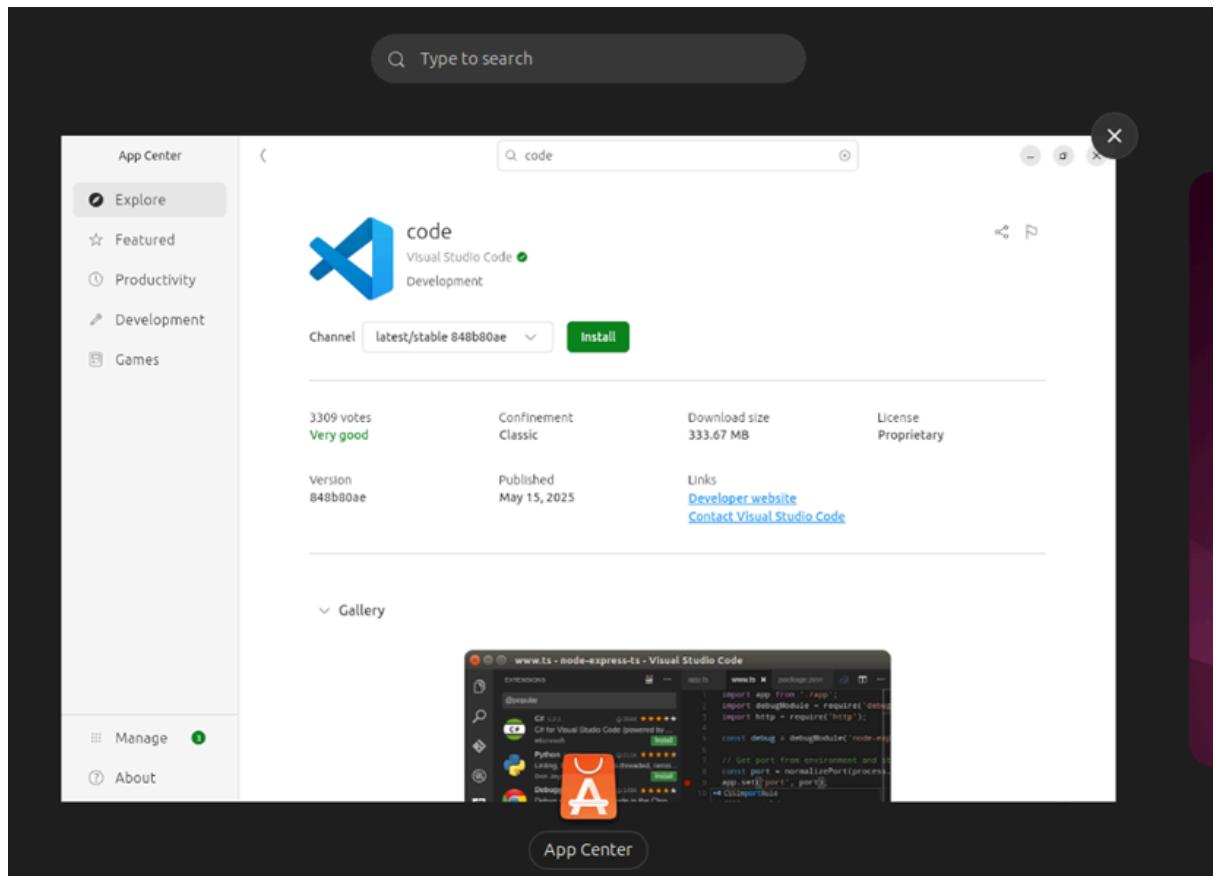


Figura 26. Captura Pantalla de la tienda App Center de Linux

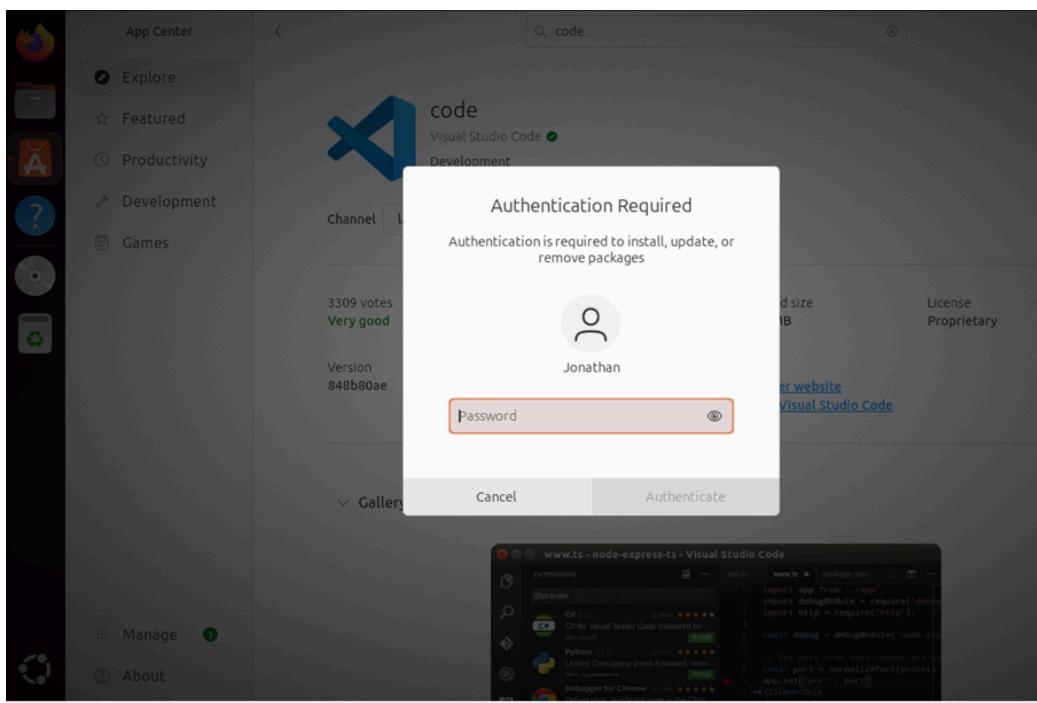


Figura 27.Captura pantalla del sistema operativo Linux, solicitud de clave de usuario.

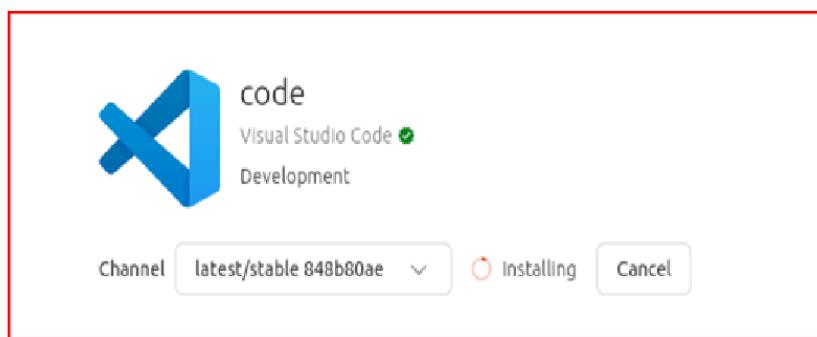


Figura 28.Captura pantalla de la tienda procesando la instalación del Visual Studio Code.

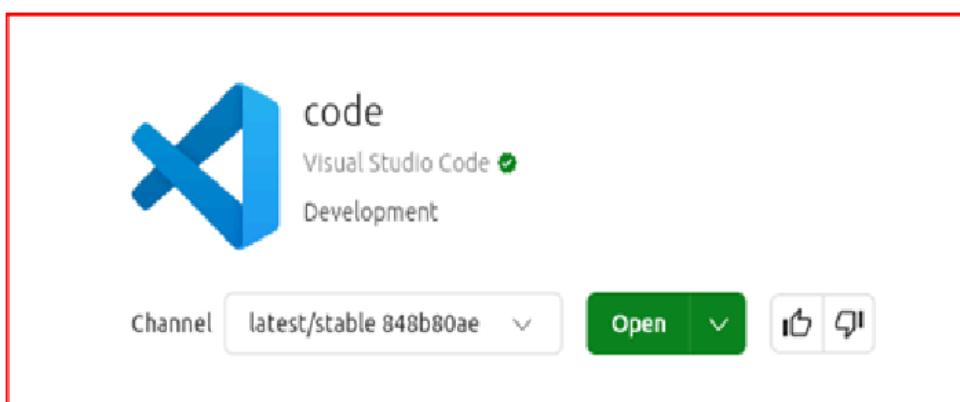


Figura 29. Captura pantalla de la tienda, descarga finalizada.

## Paso 10. Verificar si Python se encuentra instalado

Se debe ingresar a la terminal de máquina virtual, en la cual se ejecutará la línea de comando: “python3 --version” para verificar si se cuenta con alguna versión de Python.

The screenshot shows a terminal window titled "Jonathan@UBUNTU-AYSO:~". The terminal displays the following text:  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo\_root" for details.  
Jonathan@UBUNTU-AYSO:~\$ python3 --version  
Python 3.12.3  
Jonathan@UBUNTU-AYSO:~\$

Figura 30. Captura pantalla de la terminal de Linux.

## Paso 11. Descargamos la extensión de Python en Visual Studio Code

En la aplicación Visual Studio Code se procede a descargar la extensión de Python.

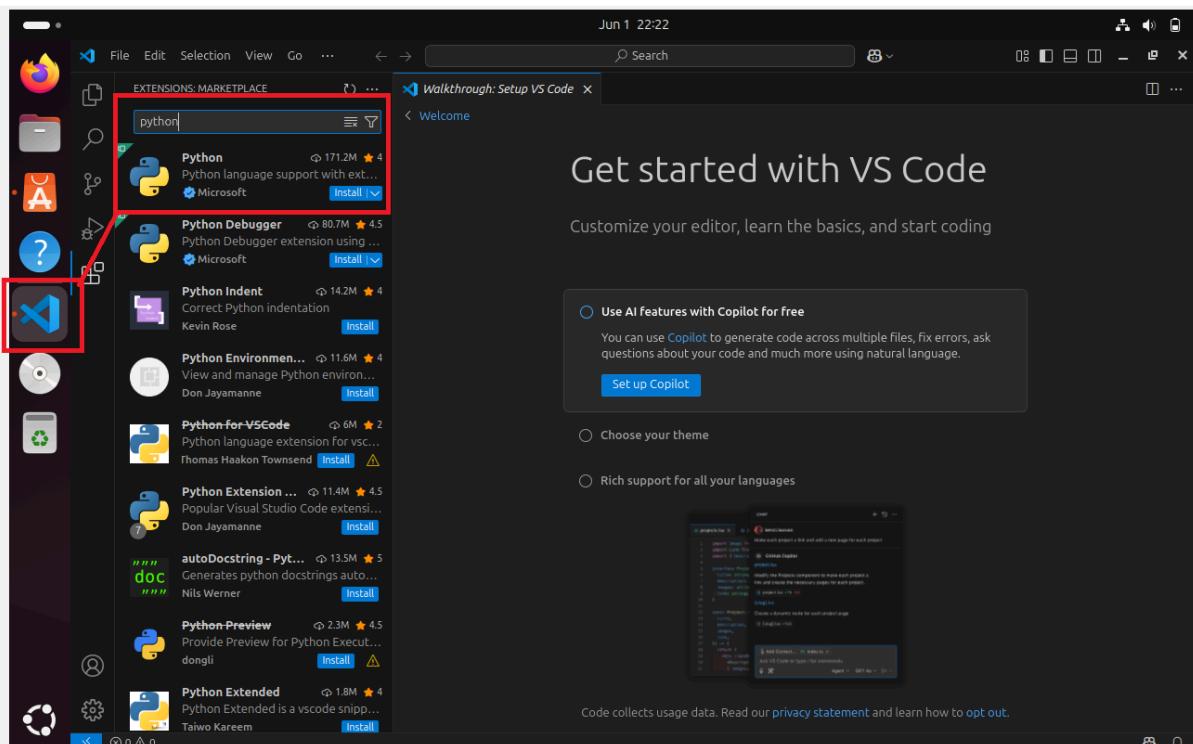


Figura 31. Captura pantalla de Visual Studio Code, descarga de la extensión de Python.

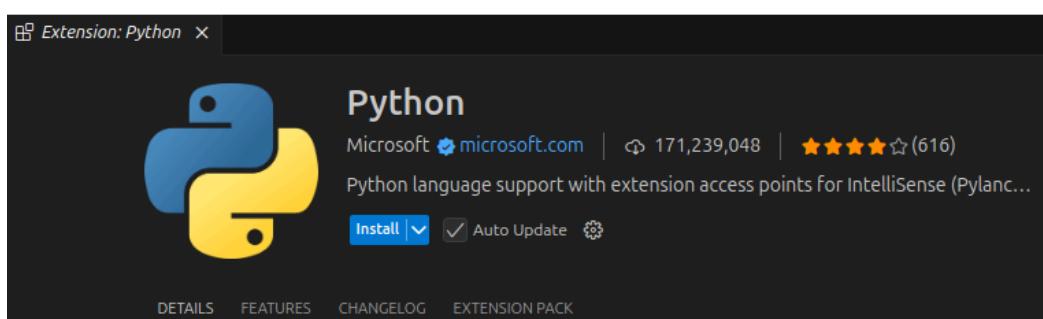


Figura 32. Captura pantalla de Visual Studio Code, descarga de la extensión de Python.

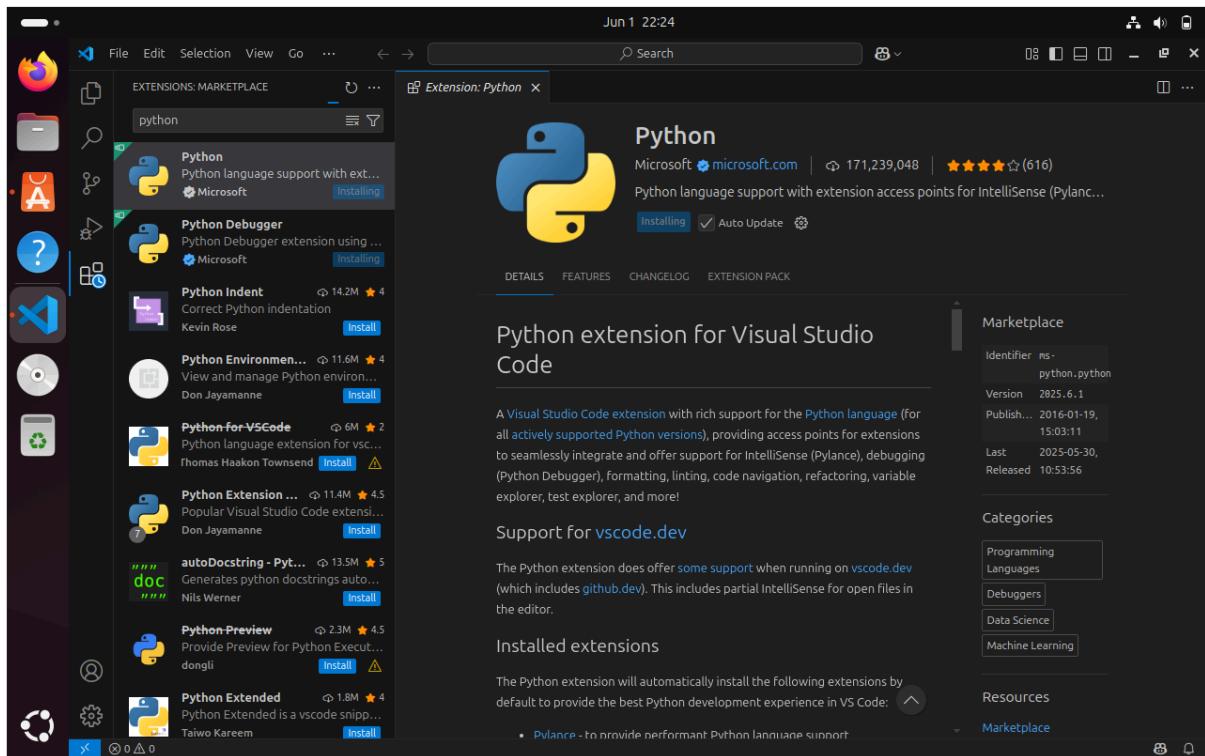


Figura 33.Captura pantalla de Visual Studio Code, descarga finalizada.

## Paso 12. Creación del archivo calculador\_promedio.py

Una vez completados los pasos anteriores, se cuenta con todo lo necesario para la creación de un archivo.py, en este caso se le coloca como nombre “calculador\_promedio”, la elección del nombre se encuentra vinculada con la finalidad del programa que se realizará.

En las siguientes capturas de pantalla se detalla el paso a paso para la creación de un nuevo archivo en Visual Studio Code.

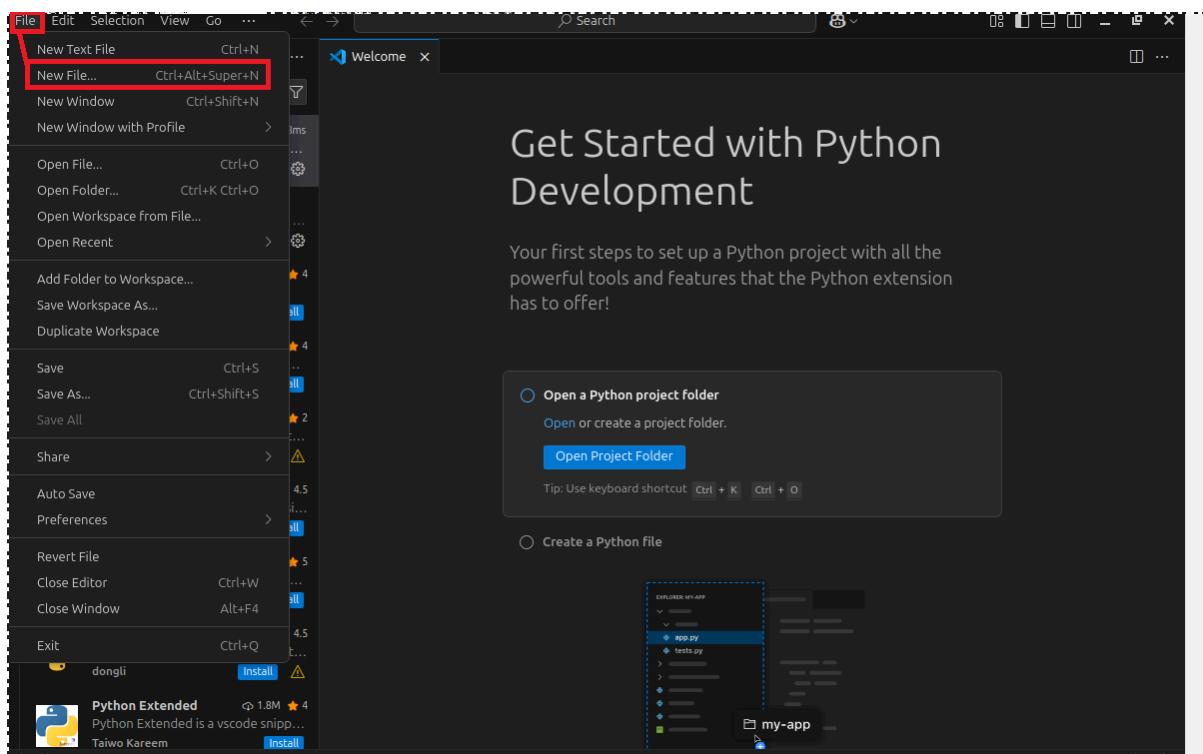


Figura 34. Captura pantalla de Visual Studio Code, creación de un archivo nuevo.

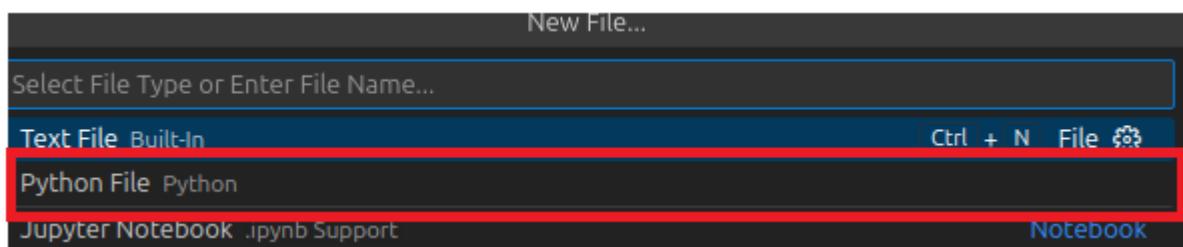


Figura 35. Captura pantalla de Visual Studio Code, creación de un archivo nuevo de Python.

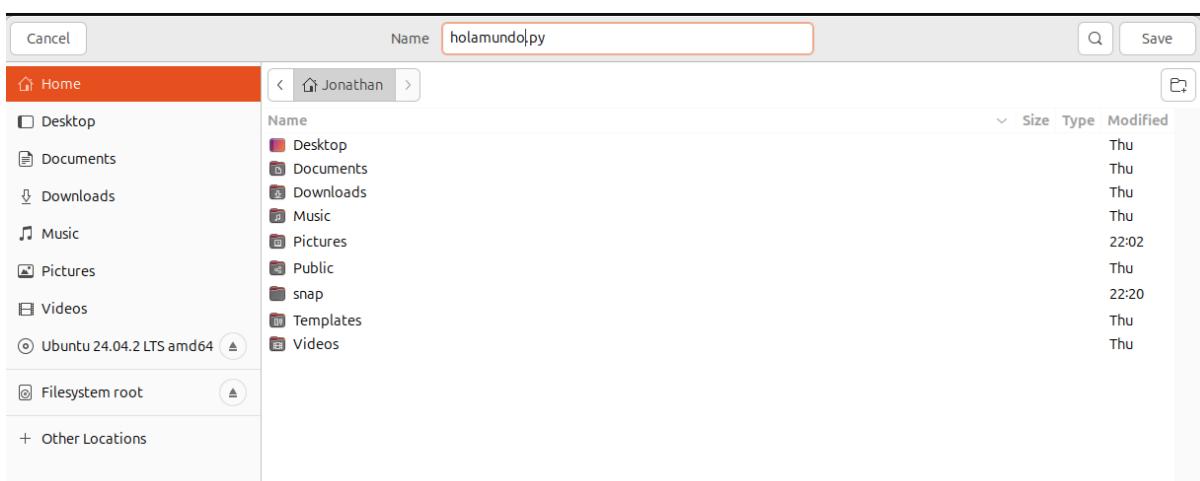


Figura 36. Captura pantalla del explorador de archivos de Linux donde se guardará el archivo.

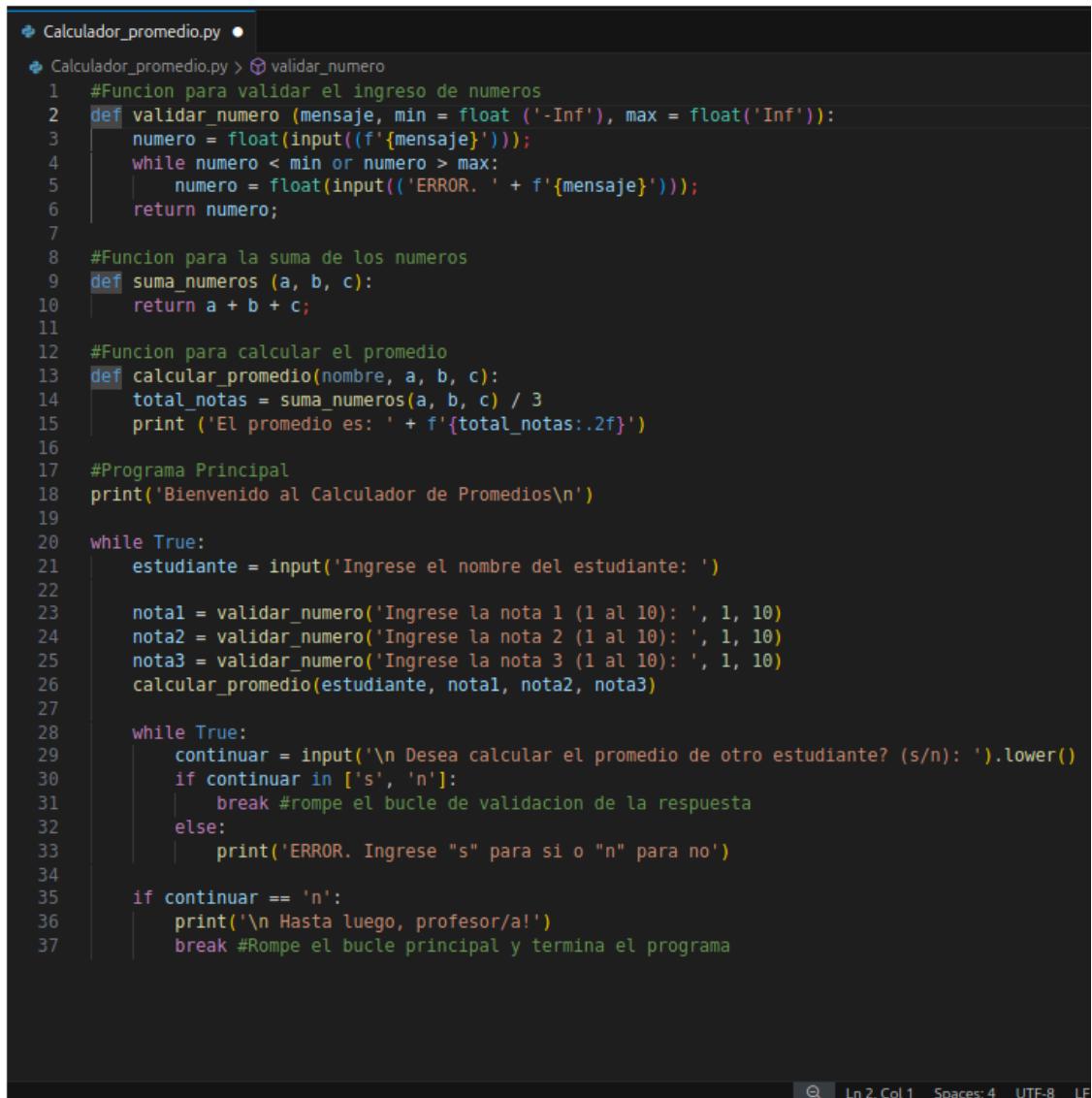
### Paso 13. Desarrollo del código.

Se procede con el desarrollo de un programa el cual se encuentra diseñado para profesores y tiene como finalidad calcular el promedio de las notas de sus estudiantes. Permite ingresar el nombre del estudiante y las tres notas correspondientes. Luego calcula el promedio y se lo muestra por pantalla. Además, le da la opción al profesor de calcular el promedio de otro estudiante o de finalizar el programa.

El código se encuentra estructurado de manera modular, lo que favorece su mantenimiento y comprensión, se han implementado funciones específicas que dividen las responsabilidades de cada parte del programa:

1. validar\_numero: solicita y valida la entrada de notas, asegurando que sean valores numéricos dentro del rango permitido (de 1 a 10).
2. suma\_numeros: se encarga de sumar las notas ingresadas.
3. calcular\_promedio:calcula el promedio de las tres notas y lo muestra en pantalla junto con el nombre del estudiante.

En el desarrollo del programa principal se implementa un bucle While que permite al docente procesar múltiples estudiantes de forma continua. Dentro de ese mismo bucle se implementa otro bucle While para validar la respuesta del usuario a la pregunta de si desea continuar o finalizar la ejecución del programa.

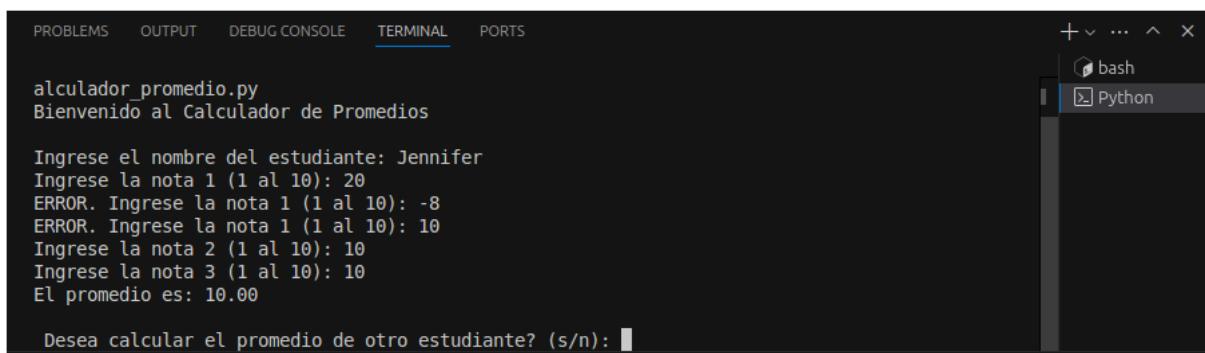


```

❸ Calculador_promedio.py •
❹ Calculador_promedio.py > validar_numero
1 #Funcion para validar el ingreso de numeros
2 def validar_numero (mensaje, min = float ('-Inf'), max = float('Inf')):
3     numero = float(input(f'{mensaje}'));
4     while numero < min or numero > max:
5         numero = float(input('ERROR. ' + f'{mensaje}'));
6     return numero;
7
8 #Funcion para la suma de los numeros
9 def suma_numeros (a, b, c):
10    return a + b + c;
11
12 #Funcion para calcular el promedio
13 def calcular_promedio(nombre, a, b, c):
14     total_notas = suma_numeros(a, b, c) / 3
15     print ('El promedio es: ' + f'{total_notas:.2f}')
16
17 #Programa Principal
18 print('Bienvenido al Calculador de Promedios\n')
19
20 while True:
21     estudiante = input('Ingrese el nombre del estudiante: ')
22
23     nota1 = validar_numero('Ingrese la nota 1 (1 al 10): ', 1, 10)
24     nota2 = validar_numero('Ingrese la nota 2 (1 al 10): ', 1, 10)
25     nota3 = validar_numero('Ingrese la nota 3 (1 al 10): ', 1, 10)
26     calcular_promedio(estudiante, nota1, nota2, nota3)
27
28     while True:
29         continuar = input('\n Desea calcular el promedio de otro estudiante? (s/n): ').lower()
30         if continuar in ['s', 'n']:
31             break #rompe el bucle de validacion de la respuesta
32         else:
33             print('ERROR. Ingrese "s" para si o "n" para no')
34
35     if continuar == 'n':
36         print('\n Hasta luego, profesor/a!')
37         break #Rompe el bucle principal y termina el programa

```

Figura 37. Captura de pantalla del programa de calcular promedio de notas.



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
+ bash
x Python

Calculador_promedio.py
Bienvenido al Calculador de Promedios

Ingrese el nombre del estudiante: Jennifer
Ingrese la nota 1 (1 al 10): 20
ERROR. Ingrese la nota 1 (1 al 10): -8
ERROR. Ingrese la nota 1 (1 al 10): 10
Ingrese la nota 2 (1 al 10): 10
Ingrese la nota 3 (1 al 10): 10
El promedio es: 10.00

Desea calcular el promedio de otro estudiante? (s/n): 

```

Figura 38. Ejemplo 1: captura de pantalla de la terminal donde se muestran varios parámetros erróneos a propósito para ver cómo se comporta el programa con respecto a los datos que ingresa el usuario.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... ^ x
/home/Jonathan/mi_entorno/bin/python /home/Jonathan/Calculador_promedio.py
● (mi_entorno) Jonathan@UBUNTU-AYS0:~$ /home/Jonathan/mi_entorno/bin/python /home/Jonathan/Calculador_promedio.py
Bienvenido al Calculador de Promedios

Ingrese el nombre del estudiante: Jennifer
Ingrese la nota 1 (1 al 10): 20
ERROR. Ingrese la nota 1 (1 al 10): -8
ERROR. Ingrese la nota 1 (1 al 10): 10
Ingrese la nota 2 (1 al 10): 10
Ingrese la nota 3 (1 al 10): 10
El promedio es: 10.00

Desea calcular el promedio de otro estudiante? (s/n): s
Ingrese el nombre del estudiante:
Ingrese la nota 1 (1 al 10): 10
Ingrese la nota 2 (1 al 10): 10
Ingrese la nota 3 (1 al 10): 10
El promedio es: 10.00

Desea calcular el promedio de otro estudiante? (s/n): n
Hasta luego, profesor/a!
○ (mi_entorno) Jonathan@UBUNTU-AYS0:~$
```

Figura 39. Ejemplo 2: captura de pantalla de la terminal donde se muestra el comportamiento del programa respecto a los datos que ingresa el usuario.

## 4. Metodología Utilizada

La metodología utilizada consta de siete etapas principales:

1. Verificación de requisitos de virtualización.
2. Instalación de Oracle VirtualBox en Windows: Descarga y configuración de VirtualBox como hipervisor tipo 2 para alojar la máquina virtual.
3. Descarga de imagen ISO oficial de Ubuntu: Descarga de la distribución Ubuntu Server desde la página oficial.
4. Creación de la máquina virtual: Configuración inicial de la máquina virtual en VirtualBox, asignación de recursos de hardware y configuración de discos duros virtuales.
5. Configuración del entorno Ubuntu: Instalación y configuración del sistema operativo Ubuntu en la máquina virtual, asegurando su correcto funcionamiento.
6. Instalación de Python: Instalación del entorno de desarrollo Python dentro de Ubuntu para poder desarrollar código.
7. Desarrollo del código para calcular promedios en Python: Codificación y pruebas de una aplicación en Python para calcular promedios.

Investigación previa

Se consultaron fuentes oficiales de documentación de Oracle VirtualBox y Ubuntu, libros especializados en sistemas operativos y virtualización, y material de estudio de la materia Arquitectura y Sistemas Operativos de la Tecnicatura en Programación de la UTN.

### Herramientas utilizadas

- Sistema Operativo Anfitrión, Windows 10/11.
- Hipervisor Tipo 2, Oracle VirtualBox.
- Imagen ISO oficial de Ubuntu Server
- Visual Studio Code con la extensión de Python.
- Lenguaje de programación, Python.

### Trabajo colaborativo

El trabajo se realizó de forma conjunta entre dos integrantes del grupo, quienes participaron activamente en todas las etapas, desde la investigación y configuración del entorno hasta el desarrollo del código y la documentación final. Compartiendo y analizando conjuntamente los resultados obtenidos, favoreciendo el aprendizaje colaborativo.

## 5. Resultados Obtenidos

Analizamos los resultados obtenidos al implementar cada una de las etapas descritas en la metodología.

Se logró instalar y configurar exitosamente Oracle VirtualBox en el sistema operativo Windows, permitiendo la creación y gestión de máquinas virtuales.

Se descargó e instaló de manera correcta la imagen ISO de Ubuntu Server, y se configuró la máquina virtual asignando memoria RAM, procesadores y disco virtual.

Una vez creada la máquina virtual, se realizó la descarga de la aplicación Visual Studio Code desde la App Server de Linux sin ningún inconveniente, luego se verificó desde la terminal si se cantaba con alguna versión de Python.

En Visual Studio Code se instaló la extensión de Python, lo que permitió configurar el entorno de desarrollo necesario para programar código. Se desarrolló y ejecutó sin problemas un programa en Python que tiene como finalidad calcular el promedio de un conjunto de notas de estudiantes introducidos por el usuario en este caso un docente.

Dicho programa fue probado en la máquina virtual, verificando su correcto funcionamiento en varias situaciones.

Los resultados obtenidos demuestran el cumplimiento de los objetivos propuestos, se logró virtualizar un entorno Ubuntu, instalar Python y desarrollar un programa que funciona de forma correcta. Para mayor detalle, en el apartado de Caso Práctico se incluyen capturas de pantalla que evidencian el correcto funcionamiento de cada etapa.

## 6. Conclusiones

El desarrollo de este trabajo práctico integrador permite comprender de manera integral el proceso de instalación y configuración de una máquina virtual con Oracle VirtualBox y Ubuntu Server, así como la configuración de un entorno de desarrollo con Visual Studio Code y Python.

Este aprendizaje resulta clave para la formación de futuros programadores, ya que facilita la adquisición de habilidades para gestionar entornos virtualizados, una herramienta cada vez más requerida en el mundo laboral actual. También, brindó la posibilidad de experimentar con sistemas operativos en un entorno controlado, sin afectar al equipo anfitrión, lo que contribuye a fortalecer competencias en administración de sistemas y configuración de entornos de desarrollo.

En nuestra experiencia, durante la instalación y el uso de la máquina virtual no se presentaron dificultades, porque previamente al caso práctico realizamos una exhaustiva investigación sobre el tema. Sin embargo, dentro de las dificultades más usuales que se podrían presentar son la falta de activación de la virtualización en el sistema, lo cual requeriría ingresar a la BIOS o UEFI y habilitar manualmente. Otro inconveniente es la asignación incorrecta de recursos (CPU, RAM o disco), lo que puede generar un rendimiento bajo en el funcionamiento de la máquina virtual y del anfitrión.

Si bien la investigación sobre virtualización fue completa se podría explorar otras herramientas de virtualización o incluso contenedores como Docker, que ofrecen alternativas interesantes para la gestión de entornos virtuales.

Además, la ejecución exitosa de un programa en Python dentro de la máquina virtual permitió poner en práctica lo aprendido, destacando la relevancia de estas herramientas para un futuro profesional en el área de programación. Sin embargo, se podrían implementar mejoras en el programa ejecutado en Python, utilizando datos complejos como listas, agregar funciones de validación de datos ingresados por el usuario.

Esta experiencia no solo permitió alcanzar los objetivos planteados en el trabajo práctico, sino que también representó una instancia de aprendizaje enriquecedora que contribuye al desarrollo de habilidades fundamentales para desempeñarse en el mercado laboral tecnológico.

## 7. Bibliografía

Muñoz, S. (2010). *Análisis de la virtualización de sistemas operativos*. Universidad de Barcelona. Recuperado de [http://www.maia.ub.es/~sergio/linked/an\\_lisis\\_de\\_la\\_virtualizaci\\_n\\_de\\_sistemas\\_operativos.pdf](http://www.maia.ub.es/~sergio/linked/an_lisis_de_la_virtualizaci_n_de_sistemas_operativos.pdf)

Oracle. (2023). *Oracle VM VirtualBox user manual (Release 7.1)*. Oracle Corporation. Recuperado de <https://www.virtualbox.org/manual/UserManual.html>

Silberschatz, A., Galvin, P. B., & Gagne, G. (2006). *Fundamentos de sistemas operativos* (7<sup>a</sup> ed.). McGraw-Hill. Recuperado de

<https://krlzjmnz.wordpress.com/wp-content/uploads/2013/06/silberschatz-fundamentos-de-sistemas-operativos-7ma-ed.pdf>

Ubuntu. (s.f.). *Ubuntu server guide*. <https://ubuntu.com/server/docs>

Boada, D. (2025, 20 de febrero). *¿Qué es Ubuntu?*

<https://www.hostinger.com/ar/tutoriales/que-es-ubuntu>

VirtualBox. (s.f.). *Documentación de VirtualBox*. <https://www.virtualbox.org/manual/>

Enferrel, A. (2025). *Virtualización - Conceptos y Beneficios*. Universidad Tecnológica Nacional.

[https://tup.sied.utn.edu.ar/pluginfile.php/9618/mod\\_label/intro/Virtualizacion-Conceptos-y-Beneficios.pdf](https://tup.sied.utn.edu.ar/pluginfile.php/9618/mod_label/intro/Virtualizacion-Conceptos-y-Beneficios.pdf)

Enferrel, A. (2025). *Virtualización de Recursos por un Hypervisor Tipo 2*. Universidad Tecnológica Nacional.

[https://tup.sied.utn.edu.ar/pluginfile.php/9618/mod\\_sos-por-un-Hypervisor-Tipo-2.pdf](https://tup.sied.utn.edu.ar/pluginfile.php/9618/mod_sos-por-un-Hypervisor-Tipo-2.pdf)

## 8. Anexo

Código del programa

A continuación, se presenta el código del programa desarrollado en Python como parte del trabajo práctico de virtualización con VirtualBox y Ubuntu. Este código permite calcular y mostrar promedios.

```
#Función para validar el ingreso de números
def validar_numero (mensaje, min= float('-Inf'), max= float('Inf')):
    numero = float(input(f'{mensaje} '));
    while numero < min or numero > max:
        numero = float(input((f'ERROR. {mensaje}')));
    return numero;
```

```
#Función para la suma de los números
def suma_numeros (a, b, c):
    return a + b + c;
```

```
#Función para calcular el promedio
def calcular_promedio(nombre, a, b, c):
    total_notas = suma_numeros(a, b, c) / 3
    print (f'El promedio es: {total_notas:.2f}')
```

```
#programa principal
print('Bienvenido al Calculador de Promedios\n')
```

```

while True:
    estudiante = input('Ingrese el nombre del estudiante: ')

    nota1 = validar_numero('Ingrese la nota 1 (1 a 10): ', 1, 10)
    nota2 = validar_numero('Ingrese la nota 2 (1 a 10): ', 1, 10)
    nota3 = validar_numero('Ingrese la nota 3 (1 a 10): ', 1, 10)

    calcular_promedio(estudiante, nota1, nota2, nota3)

    while True:
        continuar = input('¿Desea calcular el promedio de otro estudiante? (s/n): ').lower()
        if continuar in ['s', 'n']:
            break # rompe el bucle de validación de la respuesta
        else:
            print('ERROR. Ingrese "s" para sí o "n" para no.' )

    if continuar == 'n':
        print('\n¡Hasta luego, profesor/a!')
        break # rompe el bucle principal y termina el programa

```

Video explicativo

<https://youtu.be/mWMH9QvfI6k>

Repositorio GitHub

<https://github.com/JenniferFranco/tp-integrador-AYSO.git>