

## **TRABAJO PRÁCTICO INTEGRADOR**

**Virtualización con VirtualBox**

**Alumnos:**

Facundo Auciello (Comisión 24) - [facundoauciello@gmail.com](mailto:facundoauciello@gmail.com)

Ayelen Etchegoyen (Comisión 13) - [aye.etche@gmail.com](mailto:aye.etche@gmail.com)

**Materia: Arquitectura y Sistemas Operativos**

**Profesor:** Osvaldo Falabella (Comisión 13)

Tutor: Patricio Costello (Comisión 13)

**Profesor:** Osvaldo Falabella (Comisión 24)

Tutor: Gustavo Sturtz (Comisión 24)

**Fecha de Entrega:** 5 de Junio de 2025

## Introducción

Este trabajo tiene como eje central la virtualización, una tecnología que permite ejecutar múltiples sistemas operativos de forma simultánea en una misma computadora física, mediante el uso de máquinas virtuales (VM). Se eligió este tema porque es una herramienta muy utilizada en el mundo del desarrollo de software, especialmente en pruebas, entornos de aprendizaje, despliegue de aplicaciones y administración de servidores. Esto demuestra un aporte en la flexibilidad de la configuración del sistema, generando un ambiente de trabajo estandarizado que reduce problemas y facilita la entrega de software.

El trabajo busca demostrar paso a paso cómo crear una máquina virtual usando VirtualBox, instalar un sistema operativo Linux en ella (Ubuntu) y utilizar ese entorno para correr un programa en Python. De esta forma, se recorre un flujo completo de trabajo en un entorno aislado y controlado.

La virtualización es clave para la formación como técnico en programación porque permite experimentar sin riesgos, configurar entornos, optimizar los recursos del sistema y mejorar la portabilidad del software, sumando a todo lo nombrado en el primer párrafo de este trabajo.

El objetivo es que en el proceso de elaboración y conclusión de este trabajo, se comprenda no solo el uso de esta herramienta, sino también los beneficios de poder aplicarlo día a día en nuestro estudio y más adelante, en nuestro trabajo.

## Marco Teórico

### ¿Qué es la virtualización?

La virtualización es una tecnología que permite crear representaciones digitales de sistemas físicos como servidores, almacenamiento o redes. Cada entorno virtualizado se ejecuta dentro de los recursos habilitados por el equipo host, como la memoria a utilizar (RAM), la potencia del procesamiento (núcleos del procesador) y almacenamiento (capacidad de disco). Esta simulación se realiza en un entorno aislado o SANDBOX (caja segura) donde todo lo que ocurra dentro no afecta al sistema operativo ni a los datos del usuario.

Gracias a esto se ejecuta la máquina virtual (VM) en una sola computadora física. Esto optimiza los recursos y facilita tareas como pruebas, desarrollo, y administración de sistemas.

### ¿Por qué es importante la virtualización?

Los servidores físicos consumen electricidad, ocupan espacio de almacenamiento y necesitan mantenimiento. Con frecuencia el acceso a estos está limitado por la proximidad física y el diseño de la red. La virtualización resuelve todas estas limitaciones al abstraer la funcionalidad del hardware físico en el software. Es posible administrar, mantener y utilizar la infraestructura de hardware como una aplicación en la web.

### Referencias

*¿Qué es la virtualización? - Explicación de la virtualización de la computación en la nube.* (n.d.). AWS. Retrieved June 2, 2025,  
<https://aws.amazon.com/es/what-is/virtualization/>

### Beneficios de la virtualización

- EFICIENCIA: Maximiza el uso de los recursos, reduciendo el desperdicio y mejorando la utilización del hardware

- **AHORRO DE COSTOS** : la ejecución de esta infraestructura es virtual, lo que aumenta los beneficios ya que reduce la necesidad de mantener servidores físicos, costes de mantenimiento, equipo especializado y electricidad constante.
- **AGILIDAD Y VELOCIDAD**: crear y desarrollar la máquina virtual es fácil y rápido. La virtualización hace que el proceso de ejecución de entornos de desarrollo y pruebas sea mucho más rápido.
- **AUTOMATIZACIÓN Y ESTANDARIZACIÓN**: es posible definir plantillas de máquinas virtuales permitiendo duplicar la infraestructura de forma repetida y coherente evitando errores.
- **PORTABILIDAD**: proporcionan una capa de abstracción entre el software y el hardware, lo que facilita el traslado de aplicaciones a diferentes entornos.
- **ESCALABILIDAD**: permiten escalar fácilmente aplicaciones agregando más servidores virtuales para poder distribuir el trabajo entre varias máquinas virtuales. Aumenta la disponibilidad y rendimiento en las aplicaciones.
- **SEGURIDAD**: relacionado a permitir un análisis y estudio de aplicaciones dudosas, virus informáticos o pruebas de hacking, aislando de forma segura para evitar riesgos en el equipo host.

### **Máquinas virtuales e hipervisores**

Una máquina virtual es un sistema que funciona como una computadora y se ejecuta dentro de otra computadora física. Esta última se conoce como máquina *host* y las máquinas virtuales son máquinas *invitadas*.

Un hipervisor es una capa de virtualización de software permitiendo crear y ejecutar máquinas virtuales dentro de un único servidor, así como diferentes sistemas operativos.

*“El término «hipervisor», acuñado en los años 70, nació a partir de otro término que se usaba tradicionalmente para hacer referencia al kernel de un sistema operativo: «supervisor». Al usar el prefijo «iper-», el «hipervisor» se considera el supervisor de los «supervisores».”*

Stackscale (13 de Marzo de 2024). *Hipervisores: definición, tipos y soluciones.*

<https://www.stackscale.com/es/blog/hipervisores/>

## Tipos de hipervisores

### Hipervisores de tipo 1 o bare-metal (nativos):

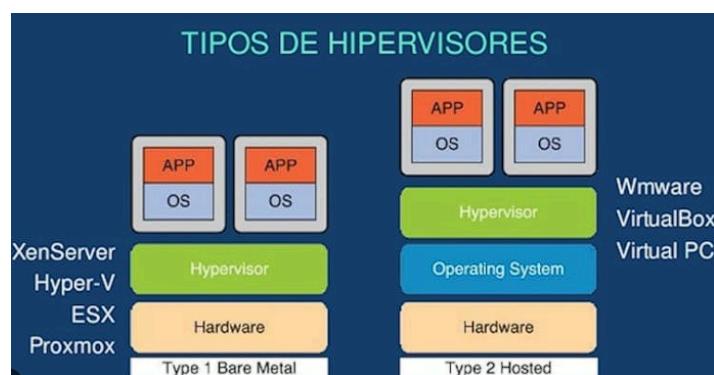
Se ejecutan directamente en el servidor gestionando el sistema principal junto con los invitados. El software de virtualización se instala directamente en el hardware.

Ejemplos: VMware ESXi, Hyper-V, Xen.

### Hipervisores de tipo 2 o alojados:

Se ejecutan como una capa de software por encima del sistema operativo del equipo host. Además pueden instalarse sistemas operativos adicionales, donde se usan para abstraer los sistemas operativos invitados del principal.

Ejemplos: VirtualBox, VMware Workstation.



La diferencia entre ambos hipervisores está en el acceso al hardware:

El Tipo 1 interactúa directamente con el hardware, ofreciendo mejor rendimiento y seguridad, ideal para servidores y entornos empresariales, pero con gasto en cantidad de servidores físicos, mantenimiento y electricidad. El Tipo 2 se ejecuta sobre un sistema operativo, lo que facilita su uso para pruebas y desarrollo, pero a costa de rendimiento y seguridad.

## Caso Práctico

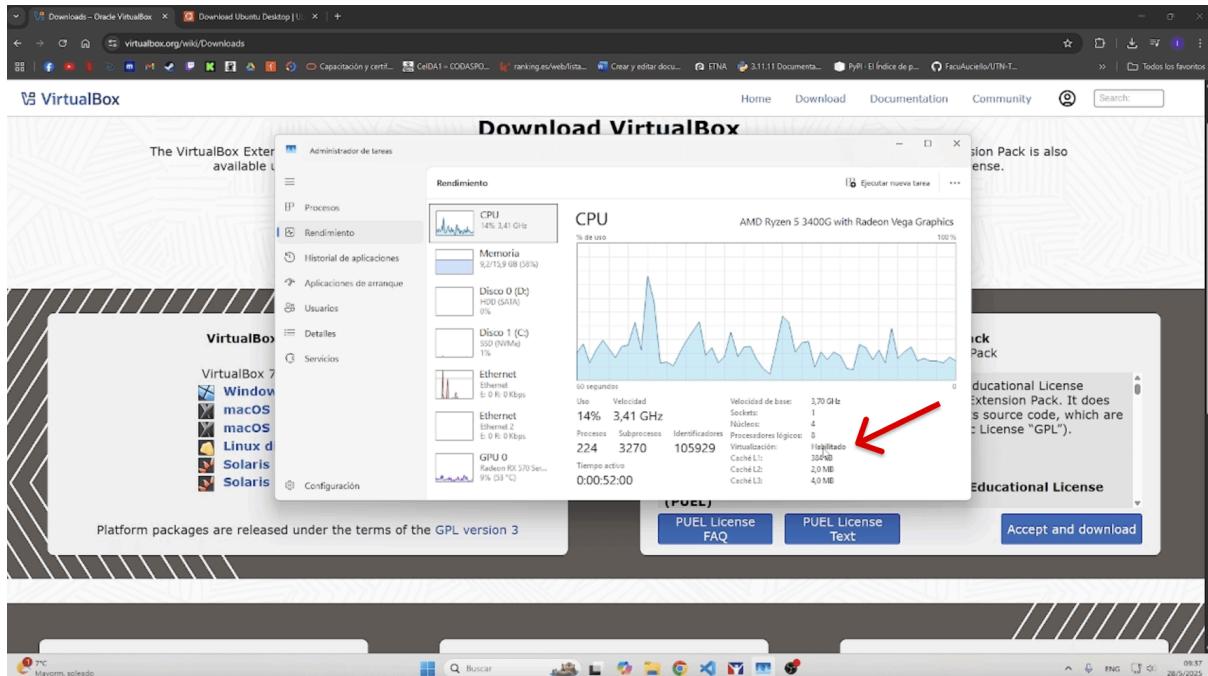
En este caso se plantea una situación en la que un usuario que tiene Windows como sistema operativo requiere de un entorno Linux para ejecutar un programa en Python. La solución a desarrollar utiliza VirtualBox para crear una máquina virtual en la que sea posible la instalación de Ubuntu y la configuración del entorno permitiéndole al usuario hacer pruebas de desarrollo con libertad en un ambiente seguro y aislado de su sistema operativo principal.

## Metodología Utilizada

Para realizar este trabajo, seguimos una serie de pasos prácticos y organizativos.

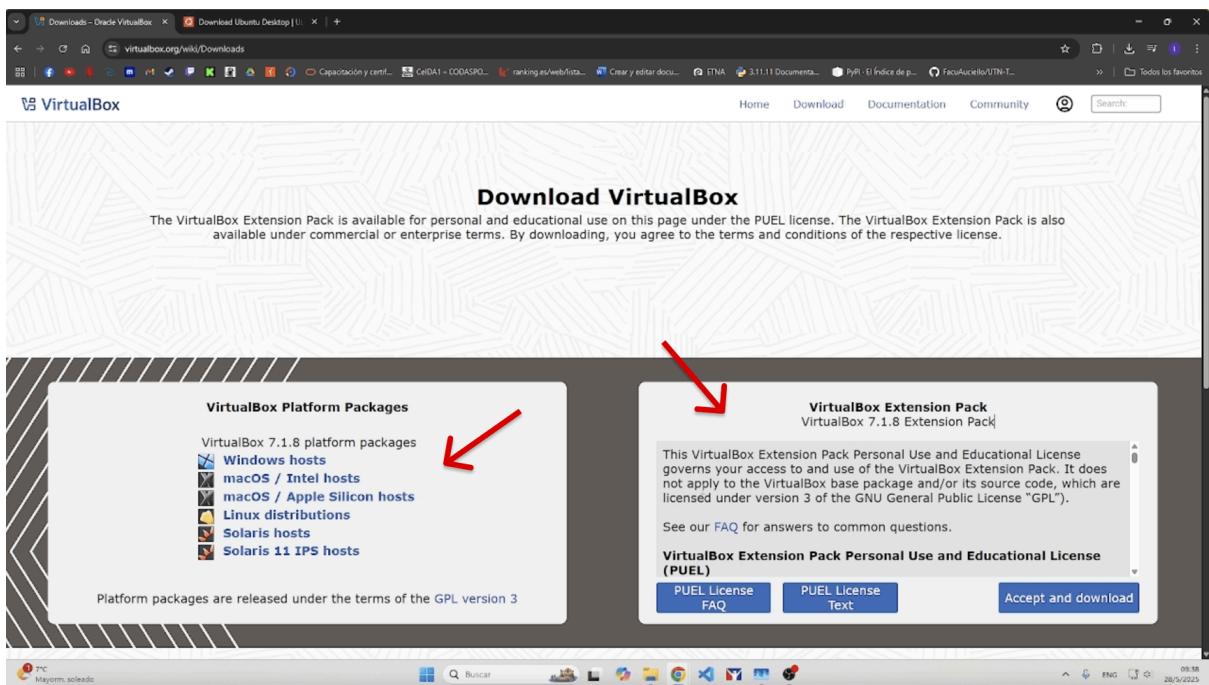
### 1. Verificación de la virtualización sobre el sistema operativo host.

Comenzamos revisando si la virtualización estaba habilitada en el sistema.



### 2. Descarga del software necesario.

Descargamos VirtualBox junto con su Extension Pack y la ISO oficial de Ubuntu Desktop.



**Download VirtualBox**

The VirtualBox Extension Pack is available for personal and educational use on this page under the PUEL license. The VirtualBox Extension Pack is also available under commercial or enterprise terms. By downloading, you agree to the terms and conditions of the respective license.

**VirtualBox Platform Packages**

- VirtualBox 7.1.8 platform packages
  - Windows hosts
  - macOS / Intel hosts
  - macOS / Apple Silicon hosts
  - Linux distributions
  - Solaris hosts
  - Solaris 11 IPS hosts

Platform packages are released under the terms of the GPL version 3

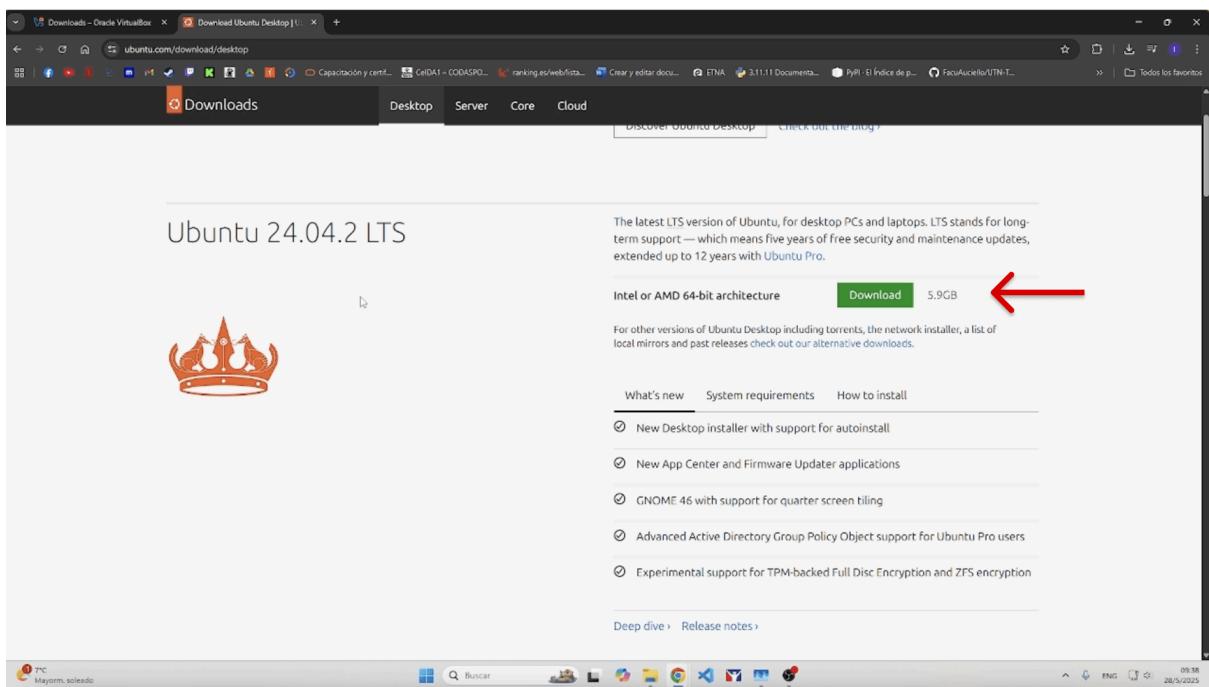
**VirtualBox Extension Pack**  
VirtualBox 7.1.8 Extension Pack

This VirtualBox Extension Pack Personal Use and Educational License governs your access to and use of the VirtualBox Extension Pack. It does not apply to the VirtualBox base package and/or its source code, which are licensed under version 3 of the GNU General Public License "GPL".

See our [FAQ](#) for answers to common questions.

**VirtualBox Extension Pack Personal Use and Educational License (PUEL)**

[PUEL License FAQ](#) [PUEL License Text](#) [Accept and download](#)



**Ubuntu 24.04.2 LTS**

The latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support—which means five years of free security and maintenance updates, extended up to 12 years with Ubuntu Pro.

**Intel or AMD 64-bit architecture** [Download](#) 5.9GB

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors and past releases check out our [alternative downloads](#).

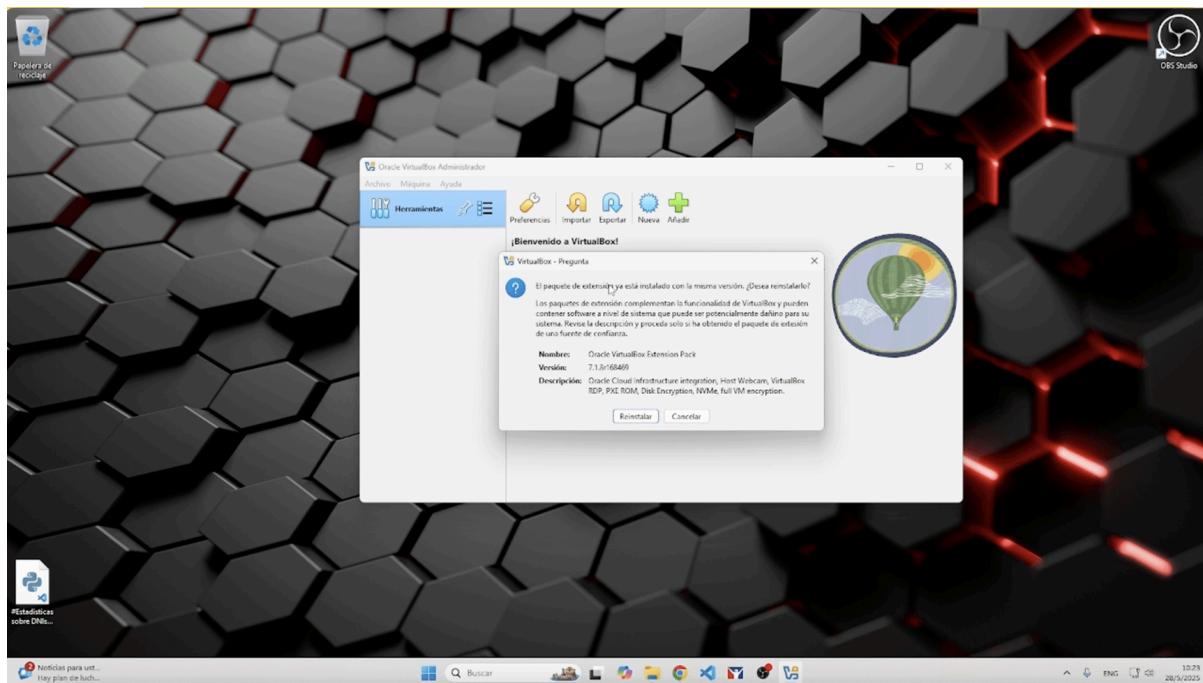
[What's new](#) [System requirements](#) [How to install](#)

New Desktop installer with support for autoinstall  
 New App Center and Firmware Updater applications  
 GNOME 46 with support for quarter screen tiling  
 Advanced Active Directory Group Policy Object support for Ubuntu Pro users  
 Experimental support for TPM-backed Full Disc Encryption and ZFS encryption

[Deep dive](#) > [Release notes](#)

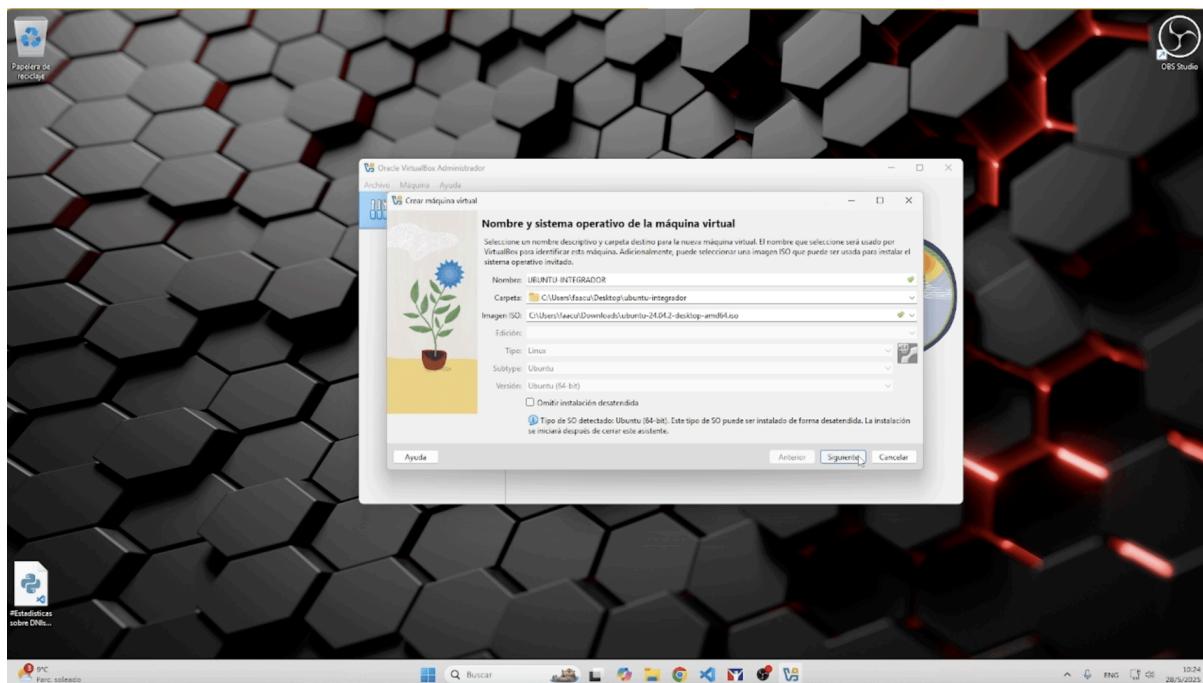
### 3. Instalación de herramientas principales.

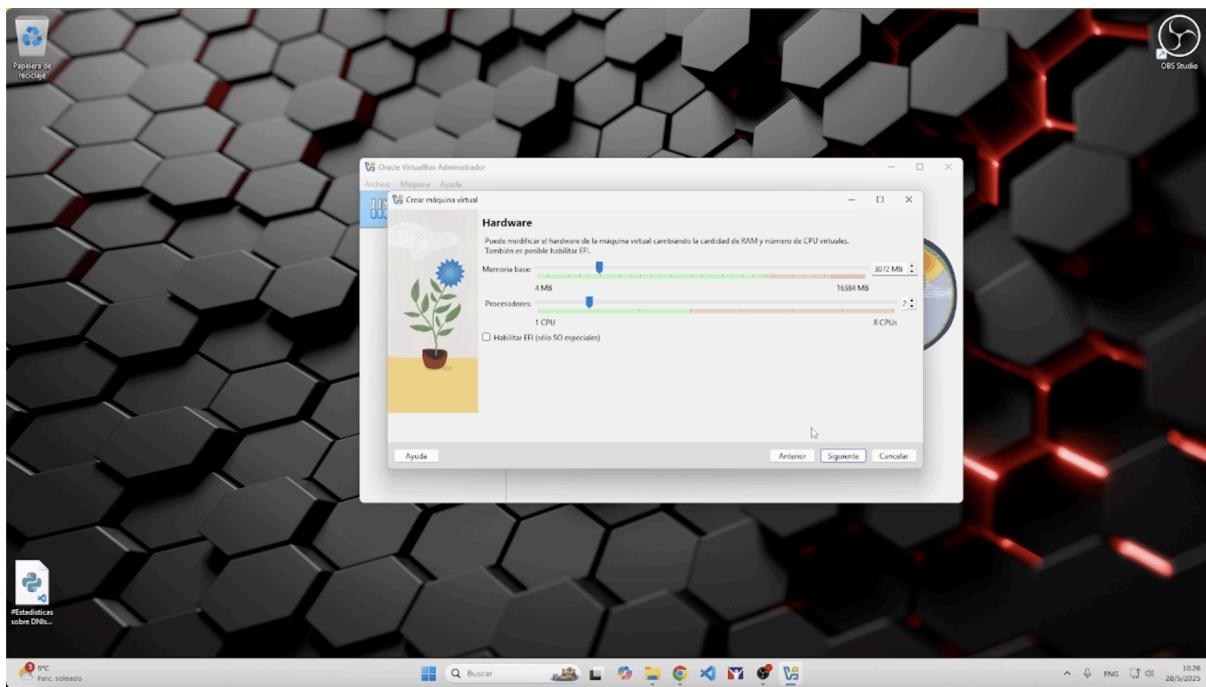
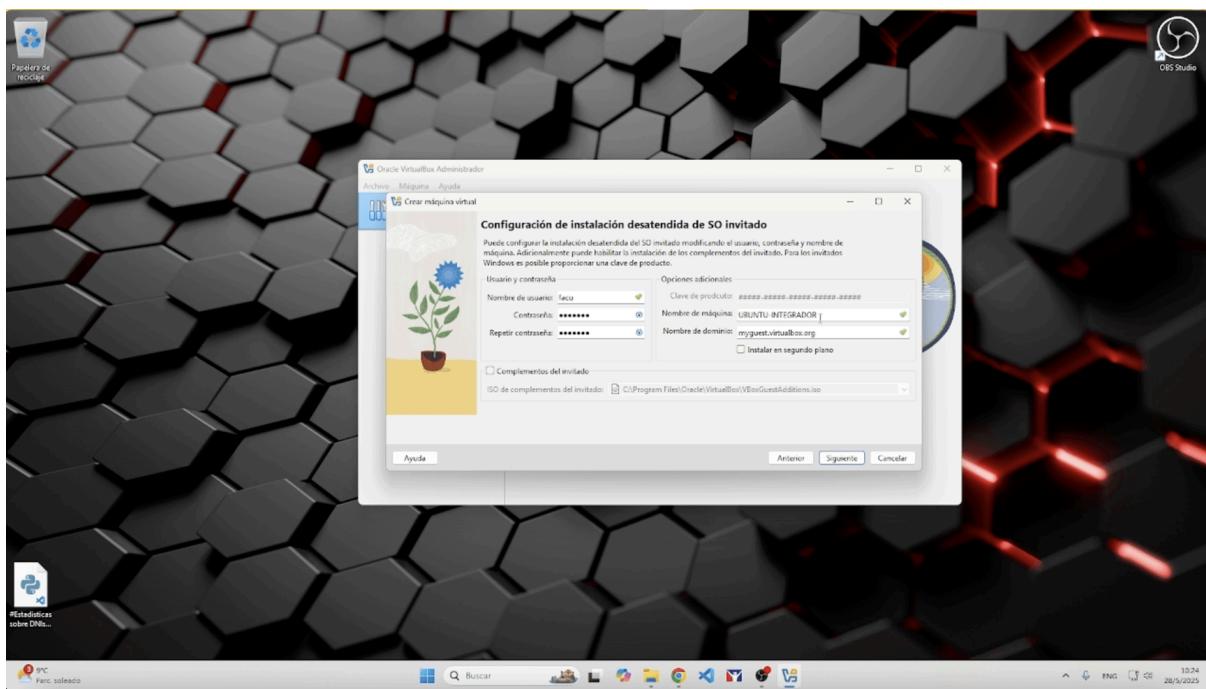
Instalamos Oracle Virtualbox y luego el Extension Pack.

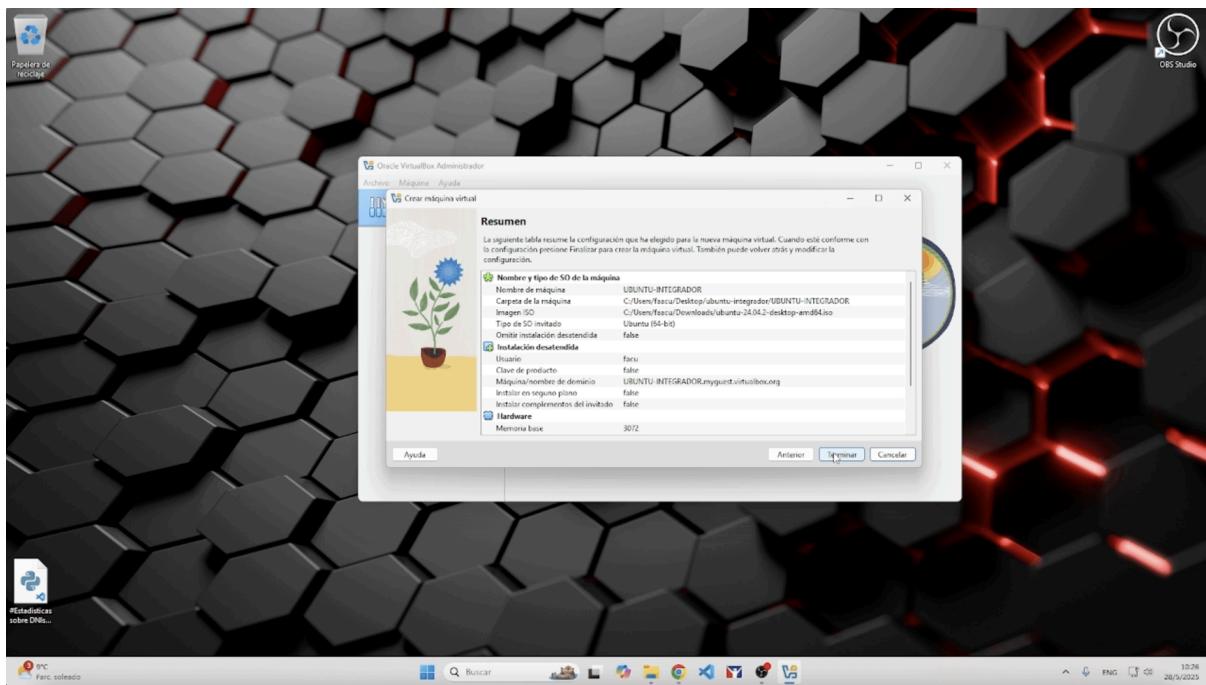
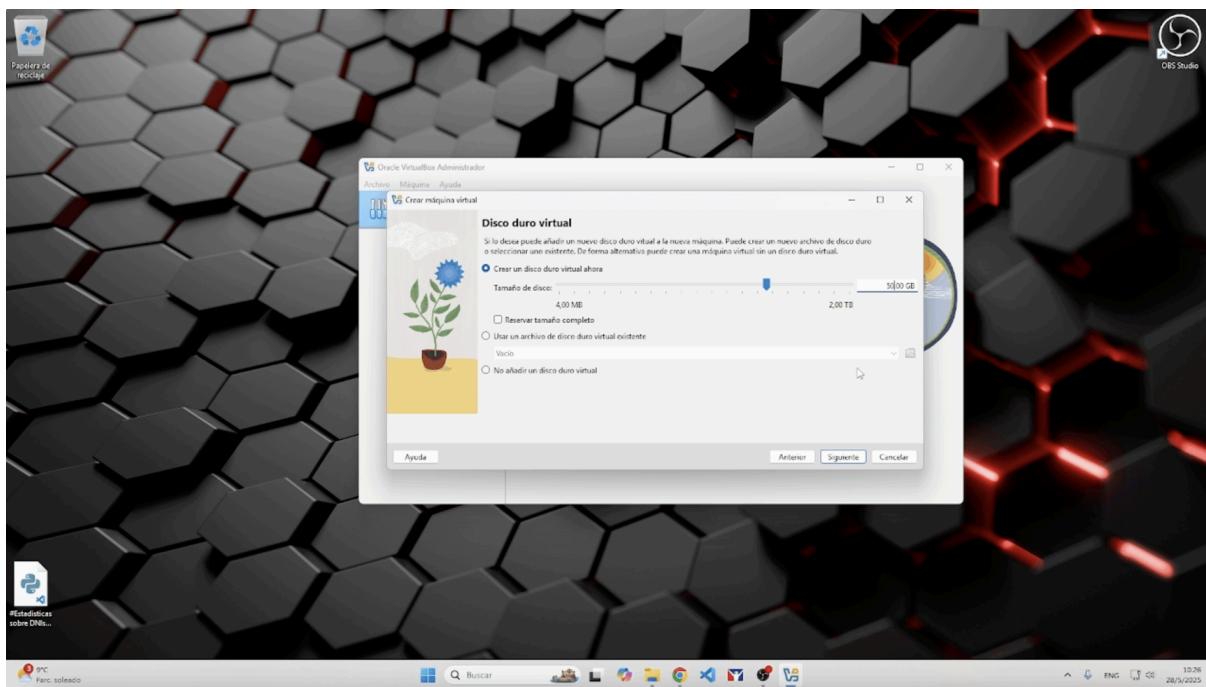


#### 4. Configuración de la máquina virtual.

Creamos una VM en VirtualBox, seleccionamos la imagen ISO de Ubuntu Desktop y configuraremos los recursos necesarios.

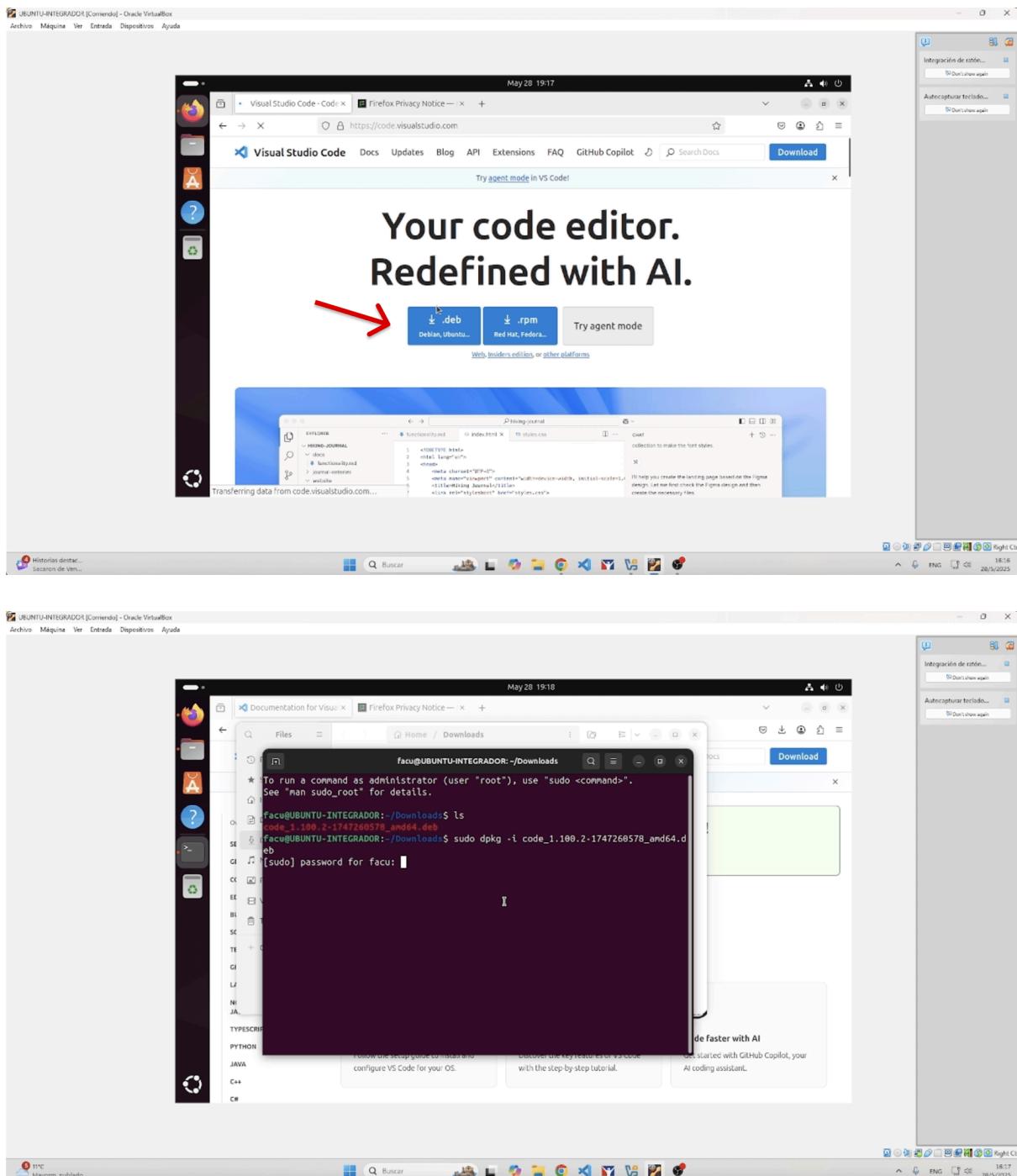






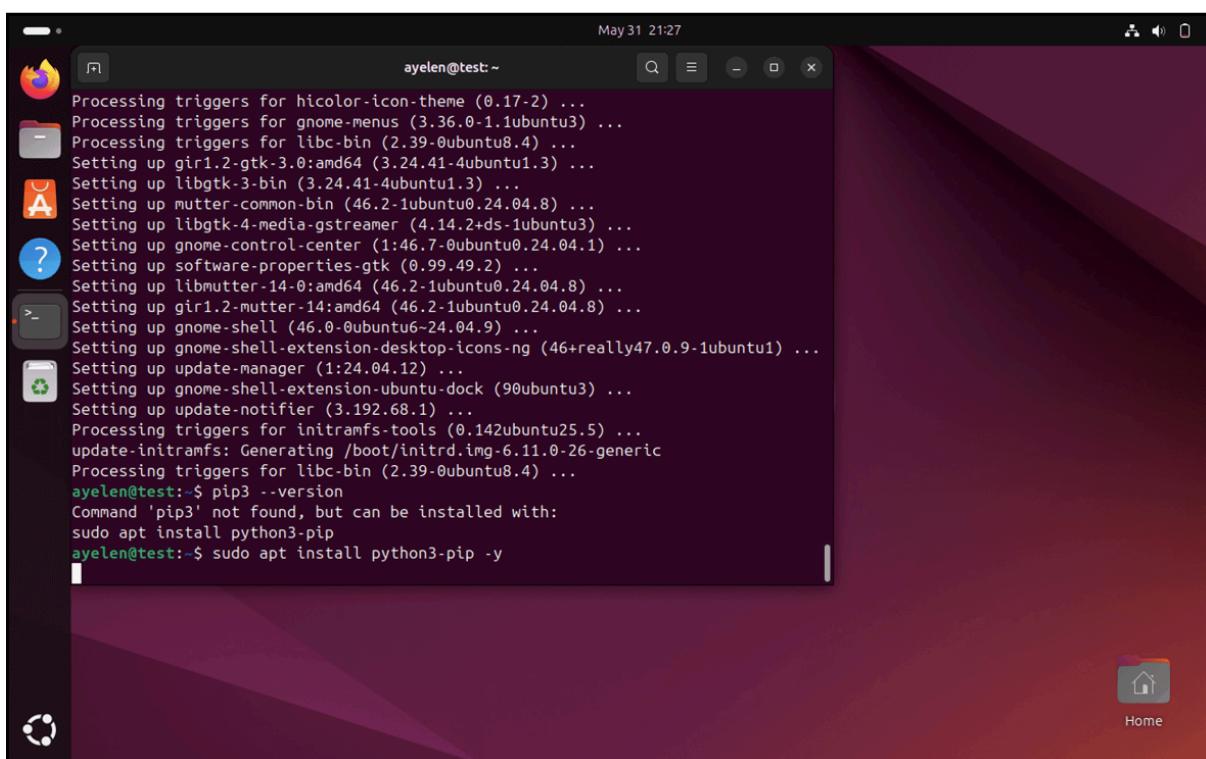
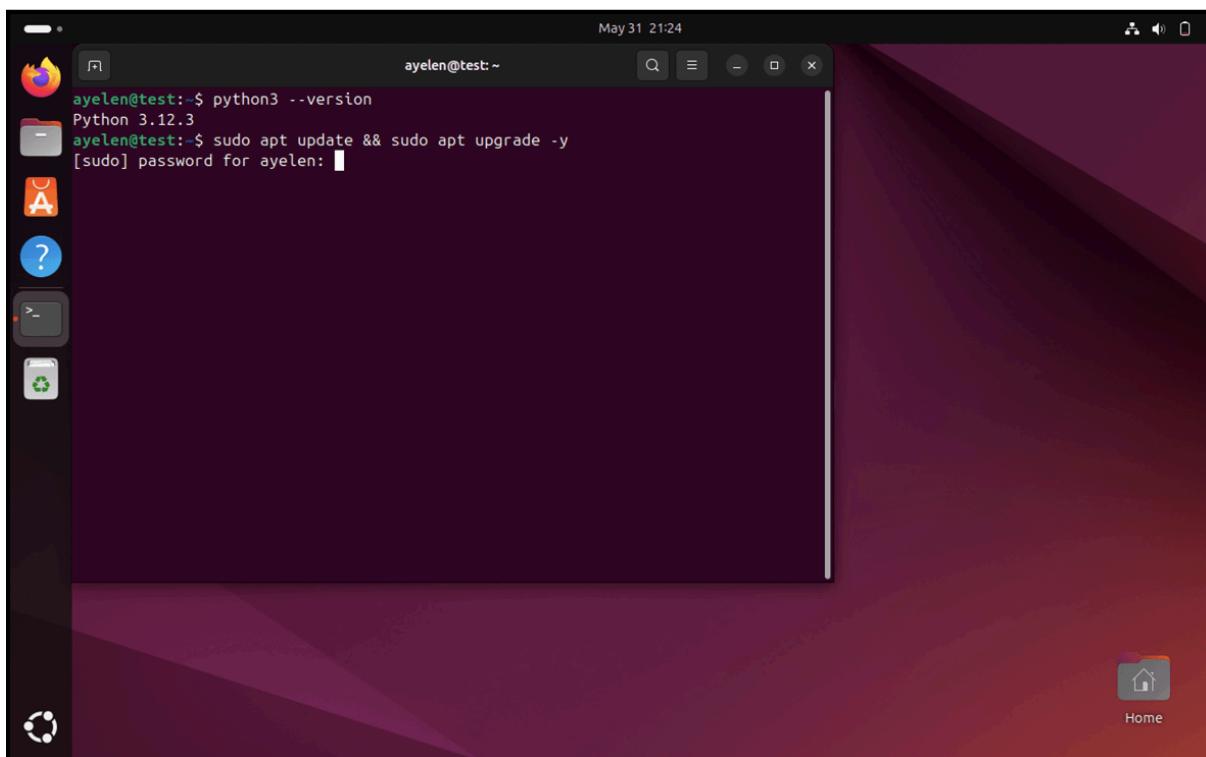
## 5. Instalación de entorno de desarrollo.

Una vez iniciado Ubuntu, descargamos e instalamos Visual Studio Code.



## 6. Instalación de Python.

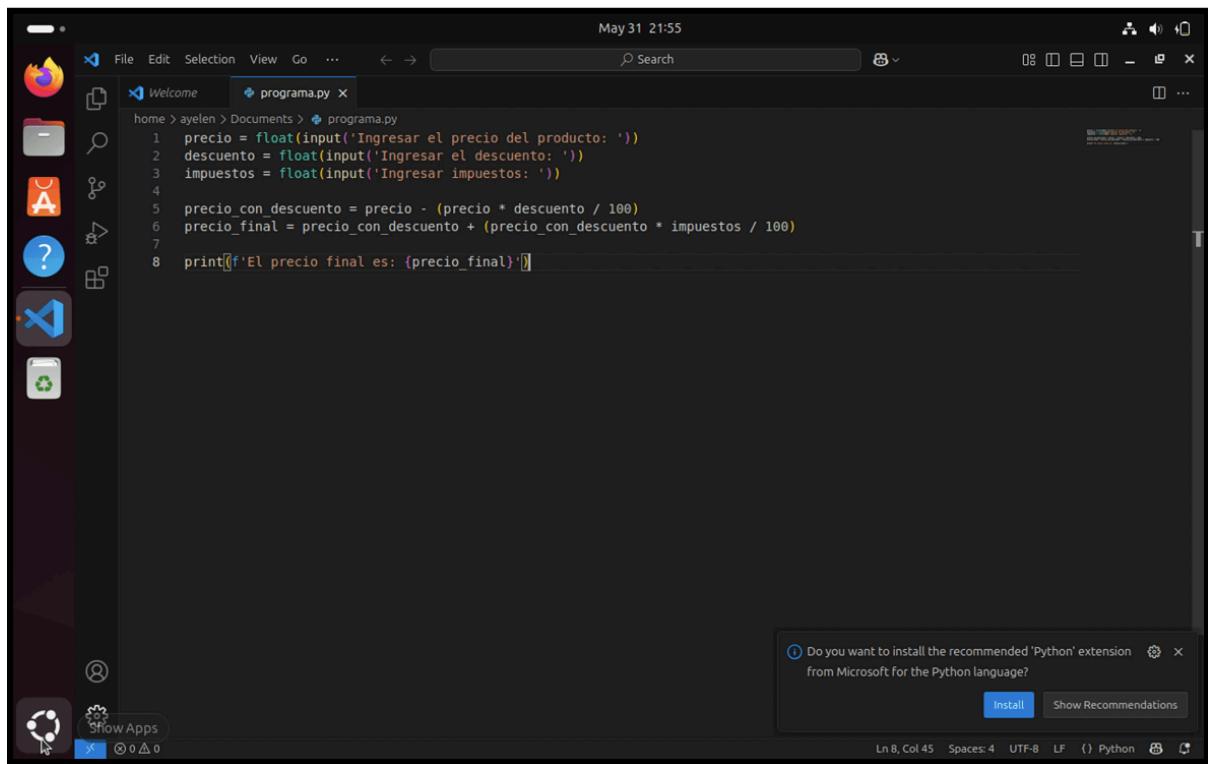
Instalamos Python 3 y Pip (gestor de paquetes).



## 7. Desarrollo del código.

Creamos un pequeño programa en Python para validar que el entorno funcione correctamente. Enlace al código del programa:

<https://github.com/FacuAuciello/TP-Integrador-AYSO/blob/main/programa.py>

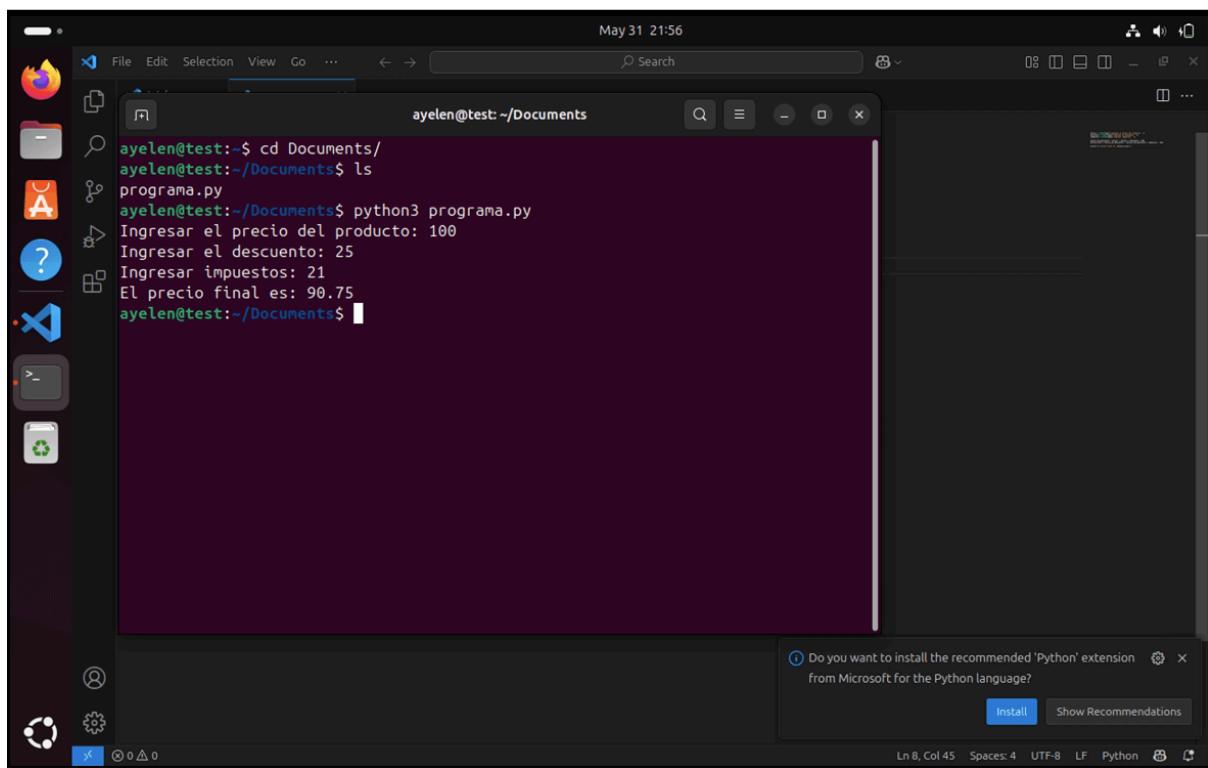


```
May 31 21:55
File Edit Selection View Go ... Search ...
Welcome programa.py
home > ayelen > Documents > programa.py
1 precio = float(input('Ingresar el precio del producto: '))
2 descuento = float(input('Ingresar el descuento: '))
3 impuestos = float(input('Ingresar impuestos: '))
4
5 precio_con_descuento = precio - (precio * descuento / 100)
6 precio_final = precio_con_descuento + (precio_con_descuento * impuestos / 100)
7
8 print(f'El precio final es: {precio_final}')


Ln 8, Col 45 Spaces: 4 UTF-8 LF Python
Do you want to install the recommended 'Python' extension from Microsoft for the Python language?
Install Show Recommendations
```

## 8. Pruebas de uso y reinicio del entorno.

Apagamos y encendemos nuevamente la VM para verificar que el sistema estuviera guardado y operativo. Ejecutamos el programa observando que todo funciona de forma correcta.



```
May 31 21:56
File Edit Selection View Go ... Search ...
ayelen@test:~/Documents$ cd Documents/
ayelen@test:~/Documents$ ls
programa.py
ayelen@test:~/Documents$ python3 programa.py
Ingresar el precio del producto: 100
Ingresar el descuento: 25
Ingresar impuestos: 21
El precio final es: 90.75
ayelen@test:~/Documents$
```

## Herramientas utilizadas

Oracle VirtualBox

<https://www.virtualbox.org/manual/>

Ubuntu desktop

<https://ubuntu.com/download/desktop>

Visual Studio Code

<https://code.visualstudio.com/>

Python

<https://www.python.org/downloads/>

## Trabajo colaborativo

El trabajo fue realizado en equipo, con una división de tareas y comunicación constante. Facundo se encargó de la primera parte del video, donde se explica qué es la virtualización, se crea la VM, se configura Ubuntu y se prueban herramientas básicas. Ayelen realizó la segunda parte del video, enfocada en la instalación de Python y versiones y el desarrollo del código dentro de la VM. Luego cierra con una conclusión.

Ambos trabajamos en la idea principal del proyecto, poniéndonos de acuerdo con el trabajo a realizar y colaborando mutuamente en parte de cada uno, además de realizar encuentros virtuales y escribir el trabajo a la par.

## Resultados Obtenidos

- Instalación y ejecución exitosa de una máquina virtual con Ubuntu como sistema operativo invitado, utilizando VirtualBox, un hipervisor de tipo 2.
- Dentro de la VM se instaló un editor de código (Visual Studio Code) y Python, desde donde se pudo desarrollar y correr un pequeño programa de prueba.
- Se verificó el funcionamiento adecuado de la virtualización, con una interacción fluida entre el sistema host (Windows) y el invitado (Linux).
- El entorno virtual funcionó de manera estable y permitió comprobar las ventajas de trabajar en un entorno aislado, ideal para pruebas y desarrollo seguro.

## Conclusiones

Durante el desarrollo del proyecto aprendimos a instalar y configurar una máquina virtual, en este caso VirtualBox, un software que permite crear entornos virtuales dentro de distintos sistemas operativos. Si bien la virtualización es un tema amplio, complejo y con muchas aplicaciones, pudimos explorar sus bases y conocer de forma práctica cómo funciona una máquina virtual. Este proceso nos permitió entender conceptos como hipervisores, entornos aislados, configuración de hardware virtual y la instalación de un sistema operativo (Ubuntu). Además, demostramos cómo trabajar dentro de una VM desarrollando un pequeño programa en Python. Esta experiencia refuerza habilidades técnicas clave para nuestra futura carrera como programadores.

En nuestro caso no se presentaron dificultades durante la instalación y el uso de la VM, sin embargo, investigamos y reconocemos algunas dificultades comunes que pueden surgir:

- La virtualización no está activada en el sistema, lo que requiere acceder a la BIOS/UEFI y habilitarla manualmente.
- Asignación incorrecta de recursos (CPU, RAM, disco), lo que puede provocar bajo rendimiento o errores en la VM.
- Problemas de compatibilidad con el sistema operativo host o con el ISO utilizado para la instalación del sistema operativo.

Llegando al final de este proyecto, pensando en futuras versiones del mismo, podríamos:

- Desarrollar un programa más completo en python agregando funcionalidades como: cálculo de cuotas en el precio, agregar validaciones, que pasa si el usuario ingresa una entrada inválida, menús interactivos, integración con archivos externos, entre otras.
- Usar un sistema operativo que sea más liviano, como Ubuntu Server, que se administra solo desde la terminal y consume menos recursos.
- Experimentar con otras herramientas de virtualización o incluso contenedores como Docker.
- Optimizar el entorno de desarrollo evitando instalar editores gráficos como VSCode si ya contamos con scripts listos.

En resumen, este trabajo nos permitió acercarnos a una tecnología fundamental en el ámbito de la programación, brindándonos herramientas para probar, desarrollar y experimentar de forma segura y controlada.

## Anexo

- Video de la presentación  
<https://youtu.be/0PJmiOYNyLQ>
- Repositorio del proyecto - README  
<https://github.com/FacuAuciello/TP-Integrador-AYSO/tree/main>
- Capturas de pantalla  
<https://github.com/FacuAuciello/TP-Integrador-AYSO/tree/main/screenshots>

## Bibliografía

- Enferrel Ariel (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 Ariel (2025), *Virtualización de Recursos por un Hypervisor Tipo 2*. Universidad Tecnológica Nacional.  
[https://tup.sied.utn.edu.ar/pluginfile.php/9618/mod\\_label/intro/Virtualizacion-de-Recursos-por-un-Hypervisor-Tipo-2.pdf](https://tup.sied.utn.edu.ar/pluginfile.php/9618/mod_label/intro/Virtualizacion-de-Recursos-por-un-Hypervisor-Tipo-2.pdf)
- Oracle (n.d.), *Oracle VirtualBox: User Guide for Release 7.1*.  
<https://www.virtualbox.org/manual/>
- B. Diego (20 de Febrero de 2025), *¿Qué es Ubuntu? Una guía rápida para principiantes*. Hostinger.  
<https://www.hostinger.com/ar/tutoriales/que-es-ubuntu>
- AWS (n.d.), *¿Qué es la virtualización? - Explicación de la virtualización de la computación en la nube*.  
<https://aws.amazon.com/es/what-is/virtualization/>
- Stackscale (13 de Marzo de 2024). *Hipervisores: definición, tipos y soluciones*.  
<https://www.stackscale.com/es/blog/hipervisores/>