

☐ Trabajo Práctico Integrado Virtualización

- **Título del trabajo:** Virtualización con Maquinas Virtuales
- **Alumnos:** (Sebastian Schneider – seba_19_sc@hotmail.com – comisión 13)
(Zair Dinahe - dinahetzair@gmail.com – comisión 4)
- **Materia:** Arquitectura y Sistemas Operativos
- **Profesor/a:** (Mauricio Pasti)
- **Fecha de Entrega:** (22/10/2025)

☐ Índice

1. Introducción
 2. Marco Teórico
 3. Caso Práctico
 4. Conclusiones
 5. Bibliografía
-

1. Introducción

Este Trabajo Integrador se enfoca en Diseñar , Configurar y Poner en Funcionamiento una Maquina virtual O Virtual Machine (VM) con un Programa escrito en Python:

- Este tema fue elegido por la relevancia que tiene el correcto uso de maquinas virtuales en el mercado laboral IT siendo una habilidad clave para el correcto desempeño de como trabajador del área
- El uso de VMs permite utilizar entornos estandarizados , seguros y homogéneos para equipos de trabajo , siendo un factor clave a la hora del Trabajo como profesionales
- Mediante es Trabajo Practico nos proponemos explicar como es la Configuración y Puesta en Funcionamiento de una Maquina Virtual , conocimientos básicos de Programación en Python y bash

2. Marco Teórico

Virtualización

- La virtualización consiste en abstraer los recursos físicos de un host para permitir la ejecución de uno o varios sistemas operativos invitados (máquinas virtuales, VM) en el mismo hardware. Este aislamiento se logra gracias a un software denominado hipervisor, que administra recursos de CPU, memoria, almacenamiento y dispositivos de E/S para cada VM, garantizando que cada sistema invitado opere como si fuera un equipo independiente.

Optimización de recursos: aprovechar mejor la capacidad del hardware físico al ejecutar múltiples entornos de forma simultánea.

- Seguridad y aislamiento: aislar cada VM del resto, de modo que un fallo o cambio en una máquina virtual no afecte al hipervisor ni a otras VMs.
- Portabilidad y escalabilidad: desplegar o migrar entornos completos (incluyendo sistema operativo y aplicaciones) de forma ágil.

Tipos de hipervisores

- **Hipervisor de Tipo 1 (bare-metal):** se instala directamente sobre el hardware, sin un sistema operativo anfitrión. Ejemplos: VMware ESXi, Microsoft Hyper-V.
- **Hipervisor de Tipo 2 (hosted):** se ejecuta sobre un sistema operativo anfitrión convencional. Ejemplos: Oracle VirtualBox, VMware Workstation.

En este TPI utilizaremos Oracle VirtualBox, un hipervisor de tipo 2 que correrá sobre Windows 11 y nos permitirá crear y gestionar la máquina virtual con Ubuntu.

Oracle VirtualBox

VirtualBox es software libre (GPL v2) que ofrece compatibilidad multiplataforma (Windows, Linux, macOS), Permite asignar CPU (núcleos), memoria RAM, espacio en disco virtual (VDI, VMDK, etc.) y dispositivos virtuales (controladoras IDE/SATA, USB, red).

3. Caso Práctico

A continuación describiremos todo el proceso de Principio a fin con Capturas de Pantallas y Descripciones de las mismas para facilitar la comprensión del mismo

Herramientas que Utilizaremos :

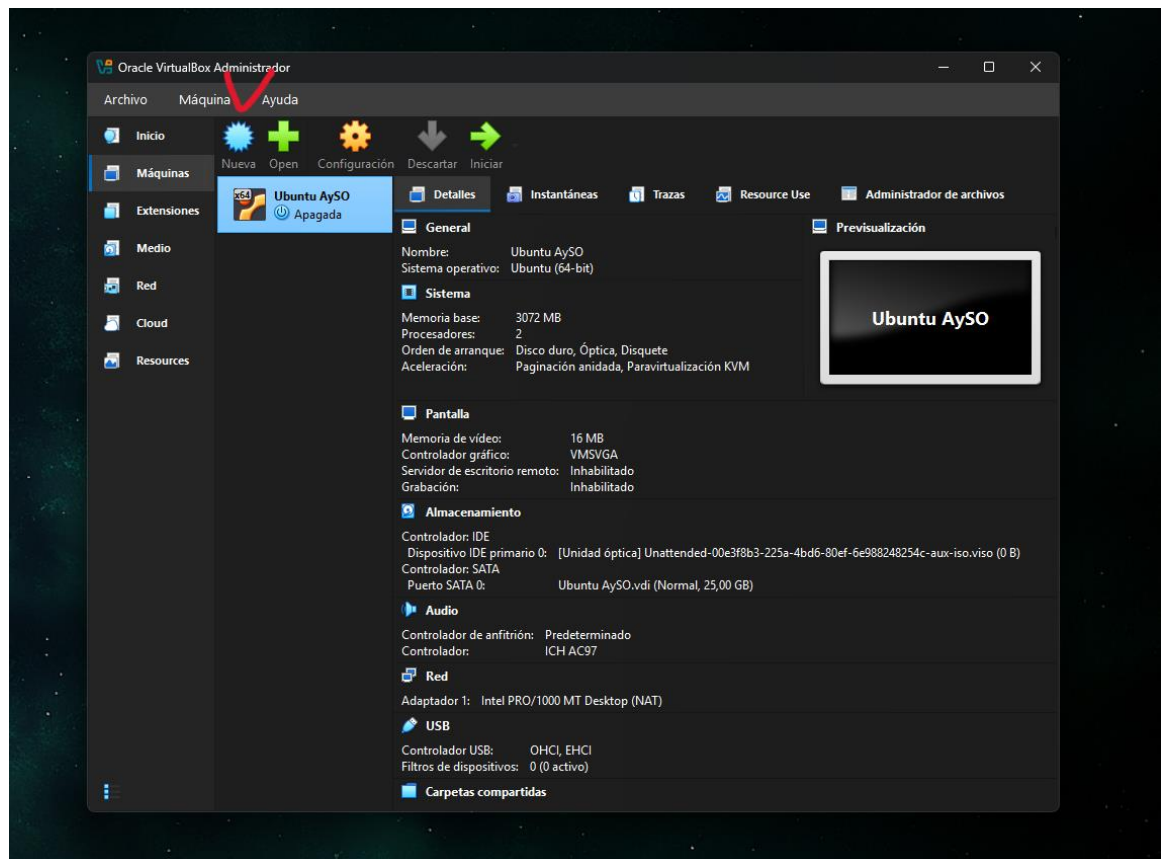
- VirtualBox de Oracle (Extension Pack)
- Ubuntu 24.04.3 LTS
- Terminal o Consola de Comandos
- Python3
- nano (editor de texto)

Paso 1 : Creación de la VM

Desde el sitio Oficial de Virtual BOX descargamos el “**Extension Pack**” :

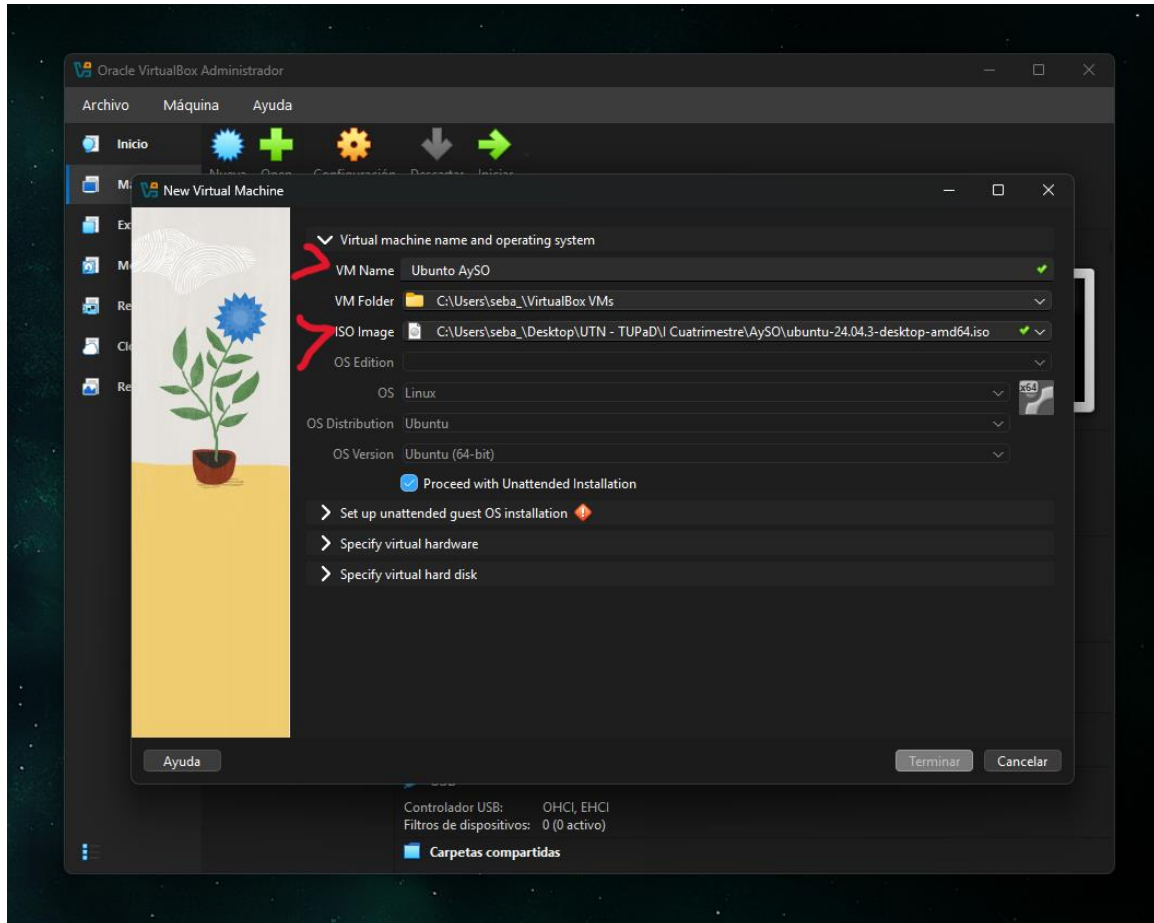
<https://www.virtualbox.org/wiki/Downloads>

Una vez Instalado lo abrimos y le damos a “Nuevo” para crear una nueva MV



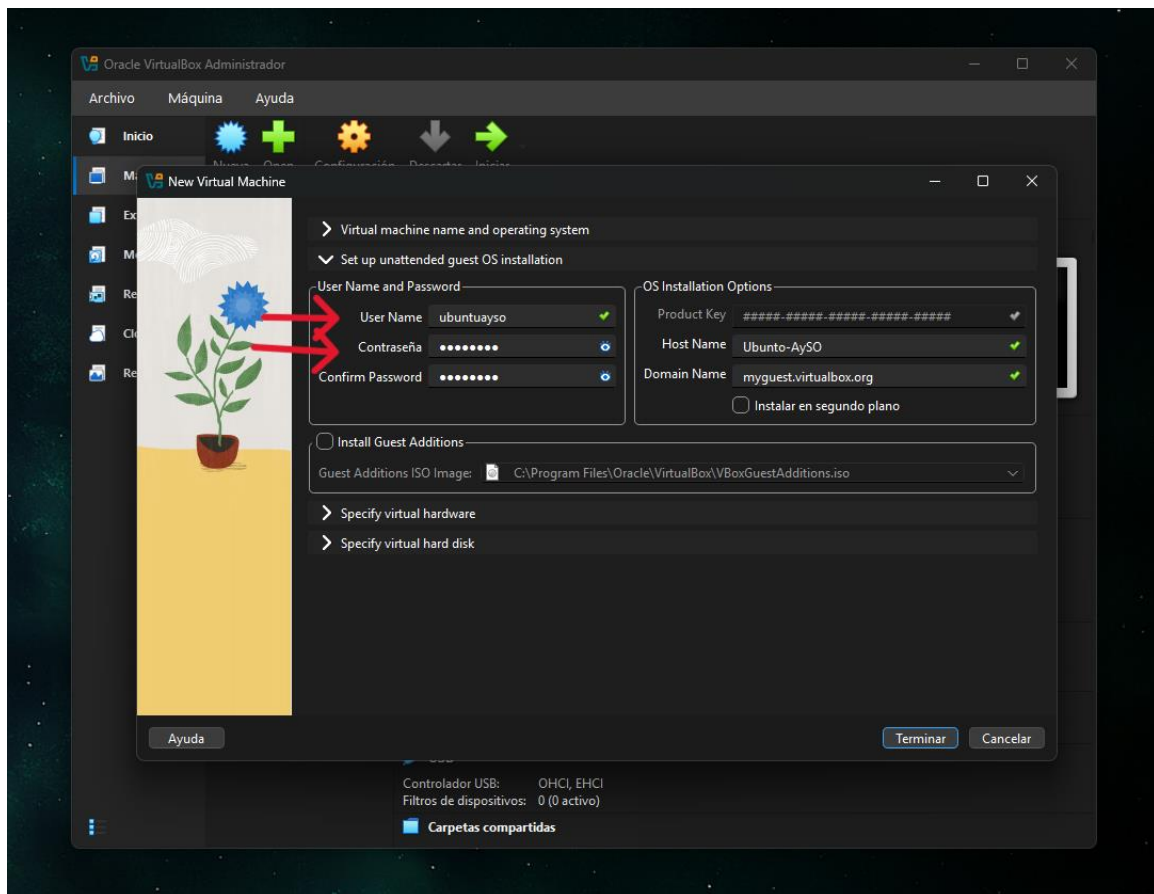
En la pestaña “Virtual Machine name” asignamos un Nombre para nuestra nueva VM y luego en “ISO Image” buscamos donde tenemos el archivo ISO del sistema Operativo que deseamos instalar , Pueden descargarlo desde aquí :

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

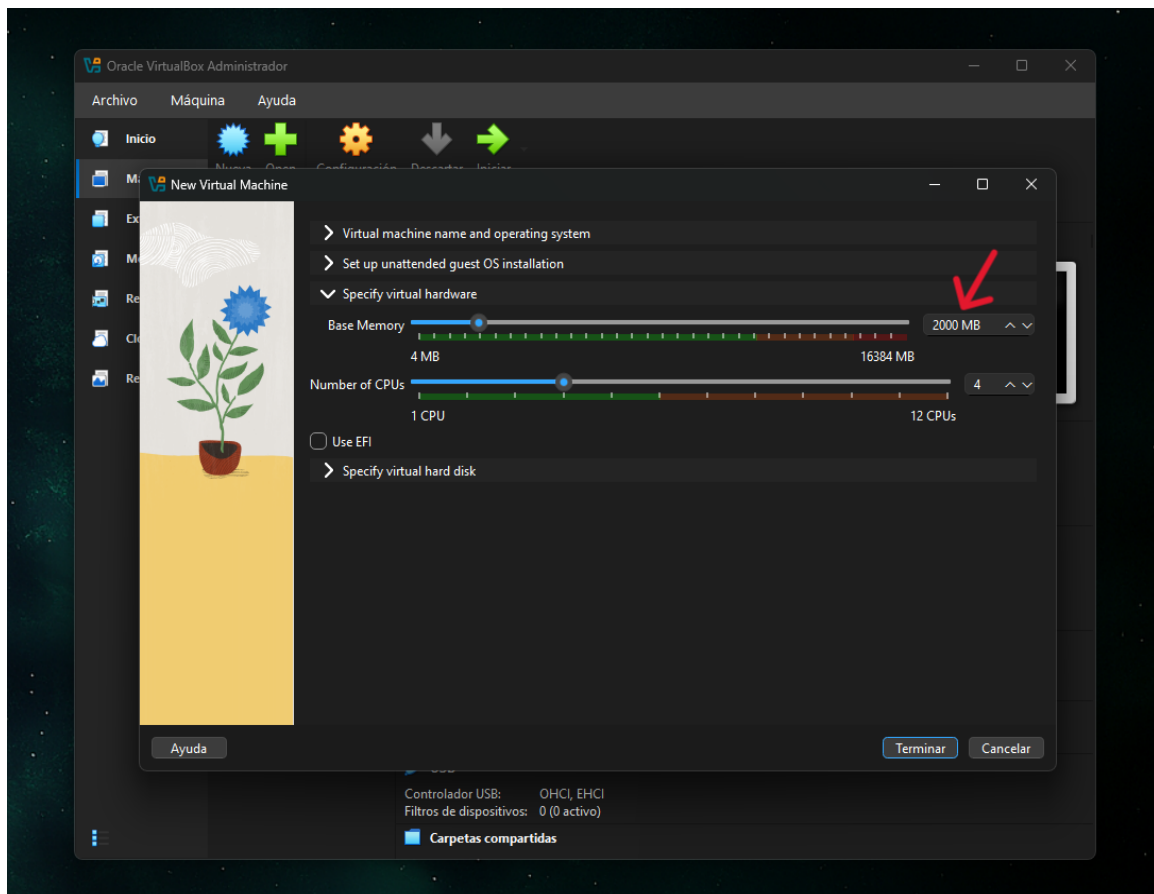


Una vez seleccionado pasamos a la pestaña “Set Up Unattended guest OS”

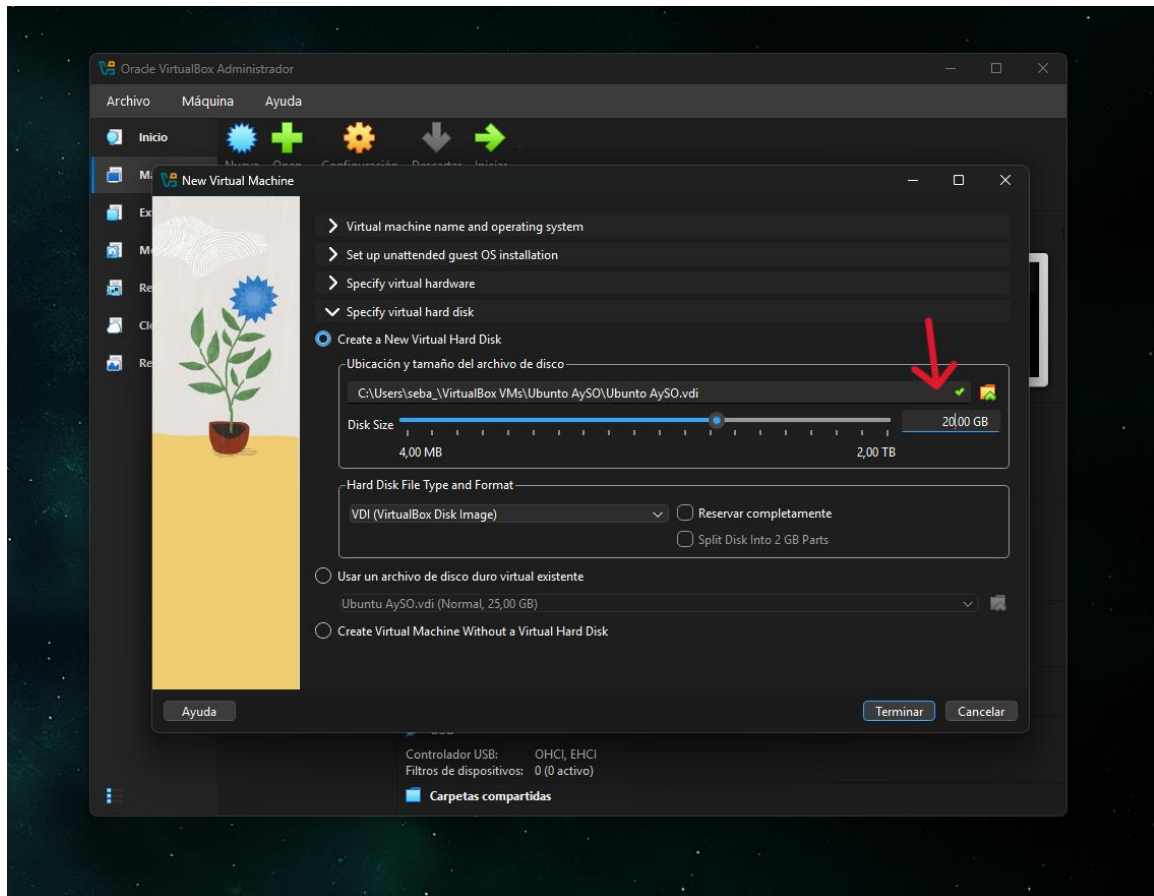
En esta pestaña asignamos Nombre de Usuario y Contraseña para luego ingresar a nuestra VM



Luego en la Pestaña “Specify virtual hardware” Indicamos la memoria que vamos a usar , en este caso son 2GB que es un mínimo para el funcionamiento del sistema

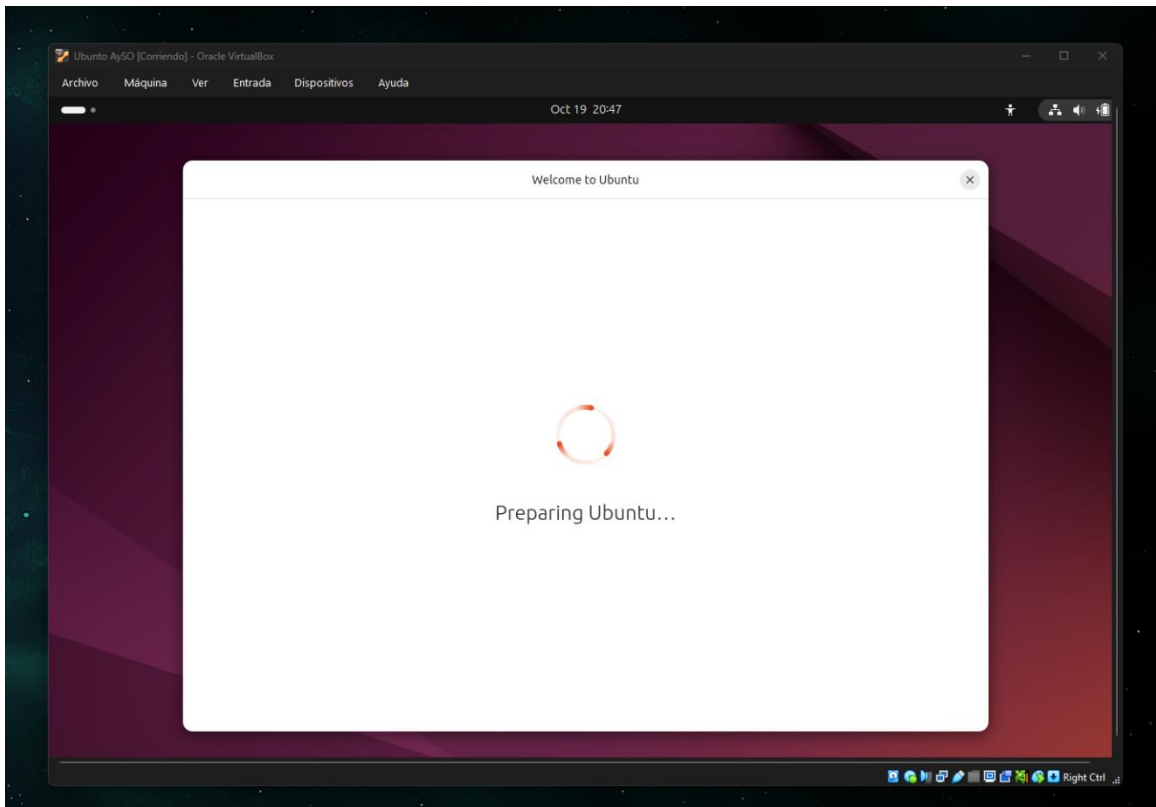


Por ultimo en la pestaña “Create a New Virtual Hard Disk” Configuramos la memoria de Almacenaje que le queremos asignar , en este caso 20GB

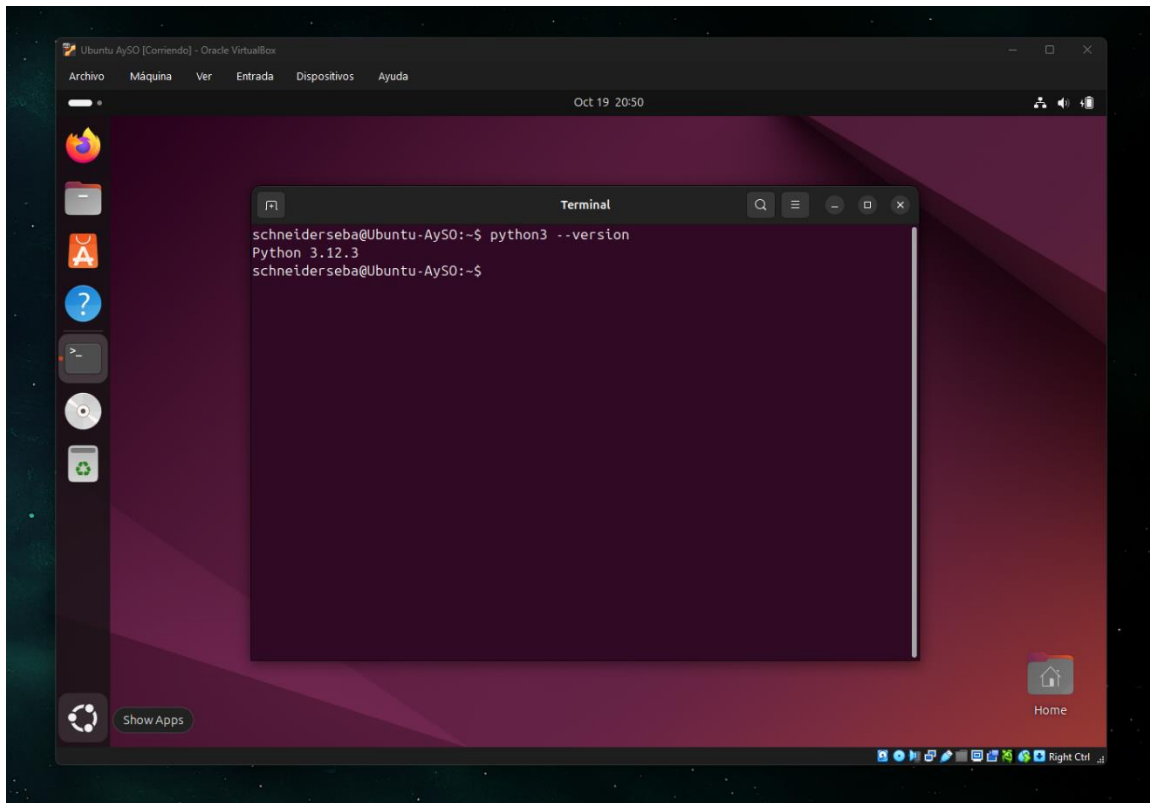


Con Estos pasos estaría Finalizada la configuración , solo quedaría darle al botón Terminar

En Este paso solo queda esperar ya que VirtualBox ejecuta toda la instalación de Ubuntu en Automático .

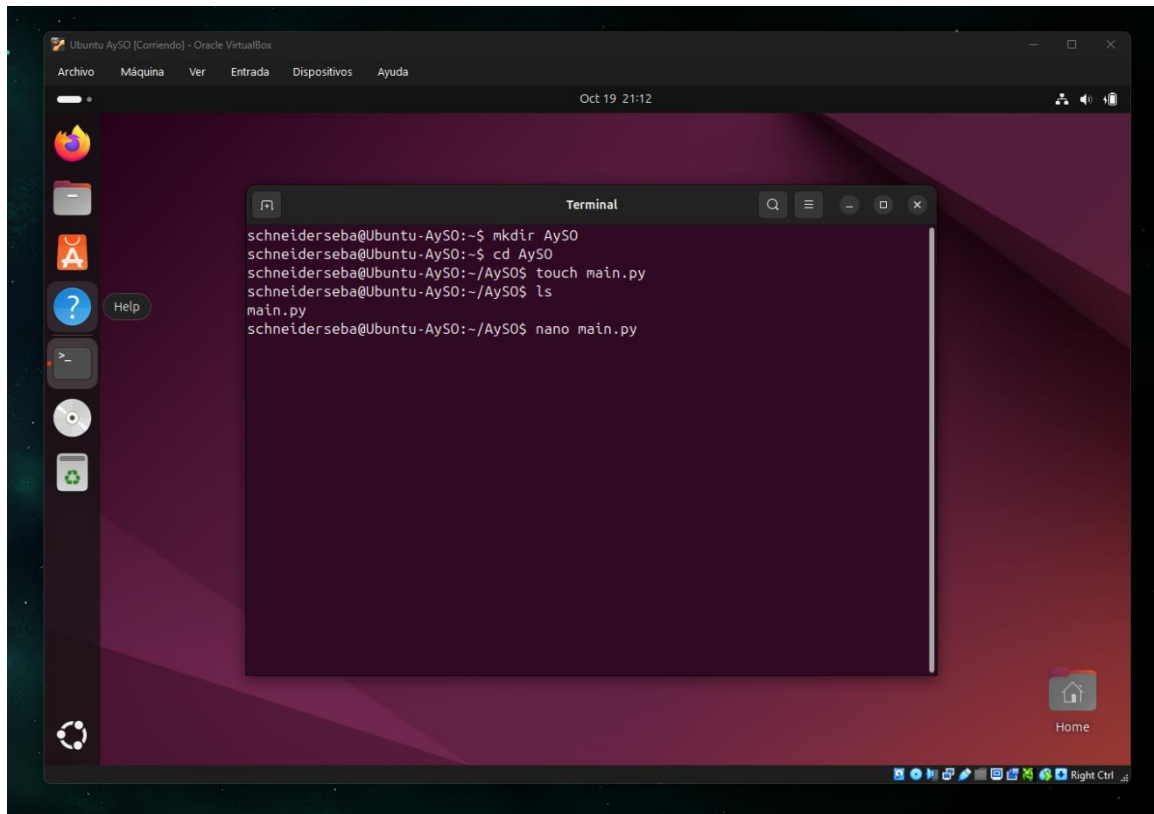


Una vez dentro del Sistema Operativo , ejecutamos la combinación de teclas Alt + Ctrl + T para abrir la Consola de Comandos en la Cual Verificamos si Python ya esta instalado o no por medio del Comando “python3 --version”

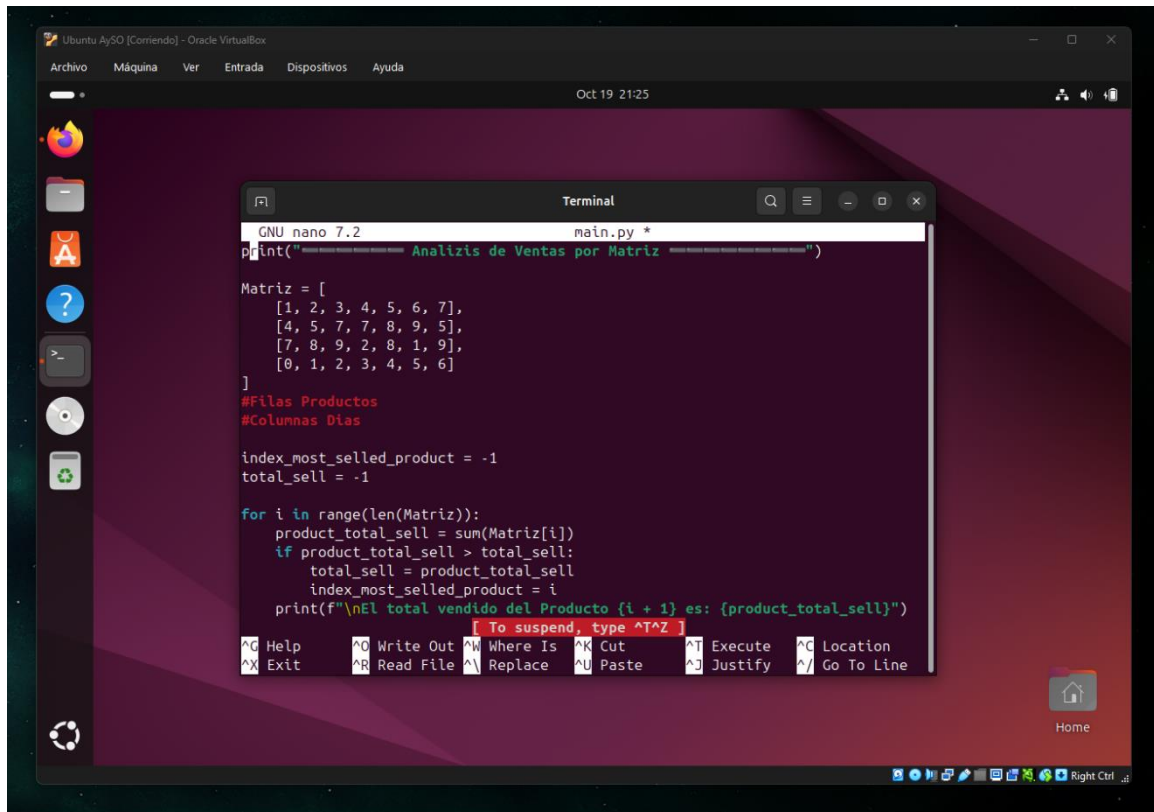


Como en este caso **Ubuntu 24.04.3 LTS** ya trae instalado Python lo siguiente que nos queda para hacer es :

- Crear el Directorio con “mkdir AySO”
- Movernos Dentro del mismo “cd AySO”
- Crear el archivo Python que vamos a ejecutar con “touch main.py”
- Por ultimo abrir ese archivo con un editor de texto para escribir el código con
“nano main.py”



Ahora Dentro del Editor Escribimos Nuestro Código , en este caso el código lo que hace es Tomar una Matriz en la cual se Encuentra las ventas de 4 diferentes ítems (Filas) en 7 días de la semana (Columnas) y en base a eso las recorre con ciclos For anidados para poder Definir diferentes Valores de Venta por ejemplo : “El día que mas se vendió” , “El Producto mas vendido y cuantas unidades” y “Total de Ventas por cada Producto”



```
GNU nano 7.2 main.py *
print("Analisis de Ventas por Matriz")

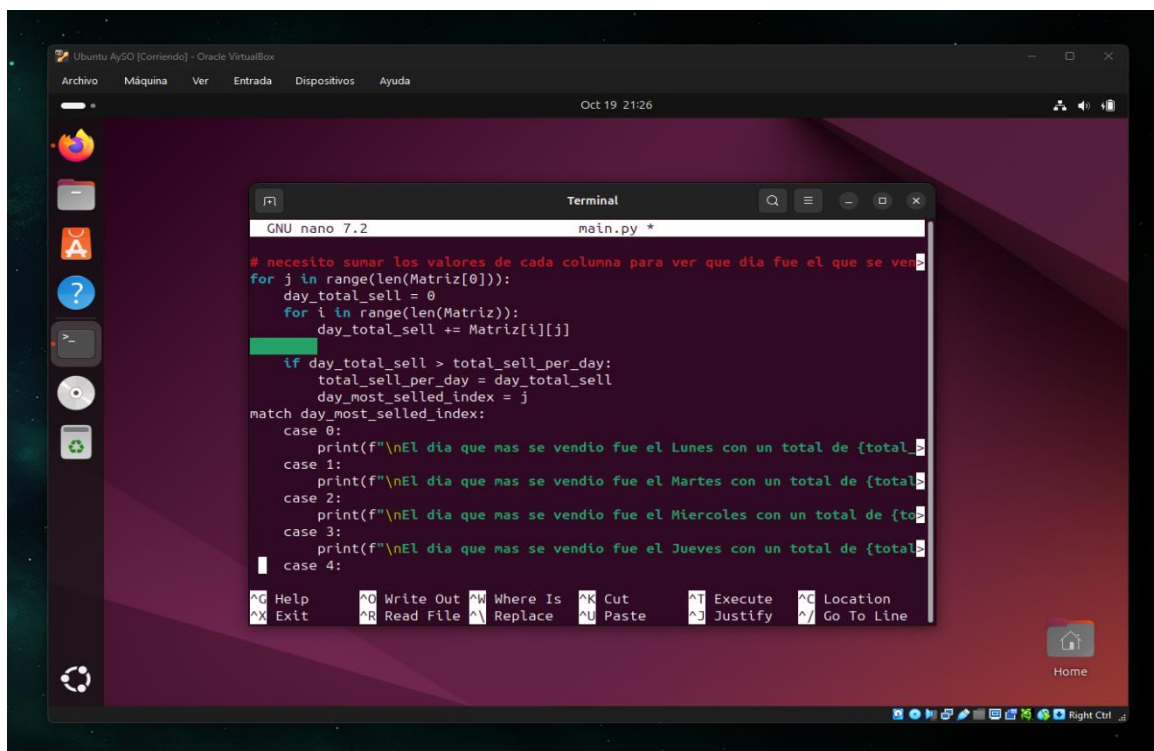
Matriz = [
    [1, 2, 3, 4, 5, 6, 7],
    [4, 5, 7, 7, 8, 9, 5],
    [7, 8, 9, 2, 8, 1, 9],
    [0, 1, 2, 3, 4, 5, 6]
]

#Filas Productos
#Columnas Dias

index_most_sold_product = -1
total_sell = -1

for i in range(len(Matriz)):
    product_total_sell = sum(Matriz[i])
    if product_total_sell > total_sell:
        total_sell = product_total_sell
        index_most_sold_product = i
    print(f"\nEl total vendido del Producto {i + 1} es: {product_total_sell}")

[ To suspend, type ^T^Z ]
^C Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

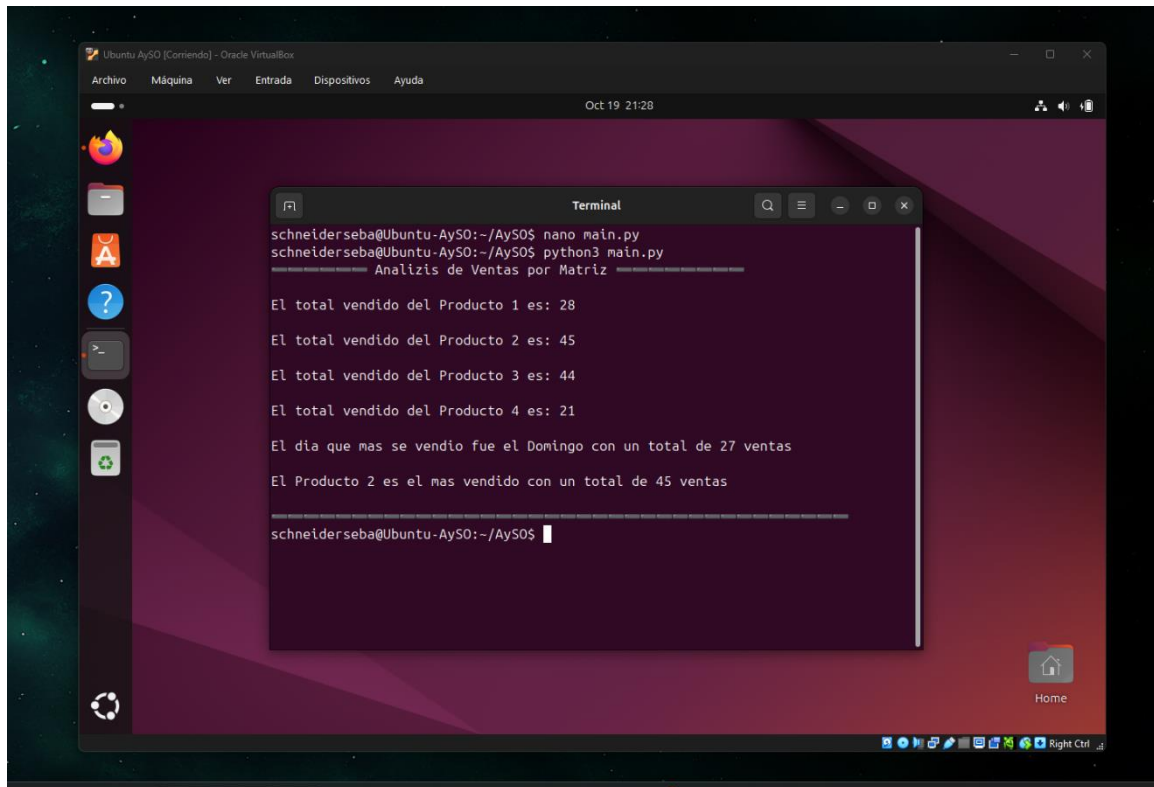


```
GNU nano 7.2 main.py *

# necesito sumar los valores de cada columna para ver que dia fue el que se ve
for j in range(len(Matriz[0])):
    day_total_sell = 0
    for i in range(len(Matriz)):
        day_total_sell += Matriz[i][j]

    if day_total_sell > total_sell_per_day:
        total_sell_per_day = day_total_sell
        day_most_sold_index = j
    match day_most_sold_index:
        case 0:
            print(f"\nEl dia que mas se vendio fue el Lunes con un total de {total_sell_per_day}")
        case 1:
            print(f"\nEl dia que mas se vendio fue el Martes con un total de {total_sell_per_day}")
        case 2:
            print(f"\nEl dia que mas se vendio fue el Miercoles con un total de {total_sell_per_day}")
        case 3:
            print(f"\nEl dia que mas se vendio fue el Jueves con un total de {total_sell_per_day}")
        case 4:
            print(f"\nEl dia que mas se vendio fue el Viernes con un total de {total_sell_per_day}")
```

Luego volvemos a la Consola de Comandos con “Ctrl + X” Y Ejecutamos el Programa con el comando “Pytjon3 main.py”



```
schneiderseba@Ubuntu-AySO:~/AySO$ nano main.py
schneiderseba@Ubuntu-AySO:~/AySO$ python3 main.py
===== Analisis de Ventas por Matriz =====

El total vendido del Producto 1 es: 28
El total vendido del Producto 2 es: 45
El total vendido del Producto 3 es: 44
El total vendido del Producto 4 es: 21
El día que mas se vendio fue el Domingo con un total de 27 ventas
El Producto 2 es el mas vendido con un total de 45 ventas

schneiderseba@Ubuntu-AySO:~/AySO$
```

Como podemos observar el Programa se ejecuta perfectamente mostrando los resultados esperados y mostrando el Correcto Funcionamiento del Entorno Virtual que creamos en Ubuntu

4. Conclusiones 🔍

En este Proyecto Pudimos adentrarnos a la Virtualización y trabajo en equipo a la vez lo que se sintió muy cercano a un trabajo en la vida real . pudimos comprender el funcionamiento de la virtualización de Tipo 2 , comprender como virtualiza sus componentes y los beneficios que ofrece , Si bien esto fue una inmersión superficial con respecto a todo lo que ofrece la Virtualización bien sabemos que esto Se podría mejorar o hacer mas simple utilizando plataformas como Google Cloud Shell que ya ofrece un ambiente listo para su uso en la nube , eliminando la necesidad de dividir los recursos de nuestra maquina , También se podría ahondar mas en la Dockerizacion con Contenedores para un mejor manejo de los ambientes de ejecución .

5. Bibliografía

Virtualización y VirtualBox:

- Material de la cátedra
- <https://www.virtualbox.org/manual>

Imagen ISO – Ubuntu 24.04.3 LTS:

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

Bash:

- Material de la cátedra.
- ChatGPT

