Informe Técnico: Implementación de Virtualización con VirtualBox y Ubuntu Desktop

Alumnos: Lerda Fernando – fglerda@gmail.com Lopez Joana – Jl105658@gmail.com

Comisión: N°16

Materia: Arquitectura y Sistemas Operativos

Profesor: Ariel Enferrel

Tutor: Emmanuel Avellaneda

Fecha de Entrega: 5 de junio de 2025

Contenido:

1. Introducción

El presente informe técnico detalla el proceso de implementación y configuración de una máquina virtual (VM) utilizando Oracle VirtualBox como hipervisor. La virtualización se ha consolidado como una tecnología fundamental en el ámbito de las Tecnologías de la Información (TI), permitiendo la optimización de recursos, mejora en la escalabilidad y la creación de entornos de desarrollo, prueba y producción aislados. Este proyecto se enfoca específicamente en la instalación de **Ubuntu Desktop 24.04.2 LTS** como sistema operativo invitado y la configuración de un servidor web **Apache2**, con el fin de proporcionar una comprensión práctica de los principios de la virtualización y sus aplicaciones directas.

Adicionalmente, se realizará una introducción comparativa entre la virtualización tradicional basada en VMs y la virtualización ligera mediante contenedores (Docker), destacando sus características y escenarios de uso. El objetivo final es documentar las metodologías, resultados y desafíos encontrados durante la configuración de este entorno virtualizado.

---

2. Historia y Evolución de la Virtualización

Las raíces de la virtualización se remontan a la década de 1960, con su implementación inicial en los mainframes de IBM, donde se utilizaba para optimizar el uso de recursos computacionales extremadamente costosos. Sin embargo, su resurgimiento y adopción masiva se produjo a finales del siglo XX y principios del XXI, impulsados por la disminución de los costos del hardware, el aumento de la capacidad de procesamiento y la creciente demanda de eficiencia en los centros de datos.

Hitos Clave en la Evolución de la Virtualización:

- **1999:** VMware lanza Workstation, democratizando la virtualización al hacerla accesible en plataformas de PC estándar.
- **2001:** Surgen los primeros hipervisores de Tipo 1 (bare-metal) como VMware ESXi, ofreciendo mayor rendimiento y eficiencia al instalarse directamente sobre el hardware.
- **2005-2006:** Intel y AMD incorporan instrucciones de virtualización (VT-x y AMD-V, respectivamente) directamente en sus arquitecturas de CPU, mejorando significativamente el rendimiento de las VMs.
- **2010s:** La virtualización se convierte en la base tecnológica fundamental para los servicios de computación en la nube pública (ej. Amazon Web Services - AWS, Microsoft Azure, Google Cloud Platform - GCP).

2.1. Marco Teórico de la Virtualización

2.2. Definición de Virtualización

La virtualización es un concepto que permite la creación de versiones lógicas o simuladas de recursos físicos computacionales, tales como servidores, almacenamiento, redes y

sistemas operativos. Su propósito primordial es maximizar la eficiencia en el uso del hardware subyacente. En lugar de dedicar una única máquina física a una tarea específica, la virtualización divide el hardware en múltiples entornos de ejecución autónomos, conocidos como Máquinas Virtuales (VMs). Cada VM opera de forma independiente, encapsulando su propio sistema operativo (SO) y aplicaciones, lo que proporciona un aislamiento significativo.

2.3. Funcionamiento del Hipervisor

El elemento central en la arquitectura de virtualización es el **hipervisor**, también denominado Monitor de Máquina Virtual (VMM). Este software actúa como una capa de abstracción entre el hardware físico del servidor y las VMs. El hipervisor es responsable de gestionar y asignar dinámicamente los recursos del hardware (CPU, RAM, almacenamiento, dispositivos de red, etc.) a cada máquina virtual según sus necesidades. Esto permite que múltiples VMs coexistan y operen simultáneamente en un mismo servidor físico sin generar conflictos, asegurando su aislamiento y la estabilidad del sistema anfitrión.

2.4. Tipos de Hipervisores

Los hipervisores se clasifican en dos categorías principales según su relación con el hardware del sistema:

2.4.1. Hipervisor Tipo 1 (Bare-Metal)

Descripción: Se instalan directamente sobre el hardware físico del servidor, sin necesidad de un sistema operativo anfitrión intermedio. Actúan como el sistema operativo principal y gestionan directamente los recursos del hardware. **Ejemplos Comunes:** VMware ESXi, Microsoft Hyper-V, Xen, KVM (Kernel-based Virtual Machine). **Características:** Ofrecen el máximo rendimiento y eficiencia al minimizar las capas de abstracción. Son la solución predilecta en entornos de producción, centros de datos empresariales y arquitecturas de nube.

2.4.2. Hipervisor Tipo 2 (Hosted)

Descripción: Funcionan como una aplicación más dentro de un sistema operativo anfitrión ya existente (ej. Windows, macOS o Linux). El hipervisor se ejecuta sobre el SO anfitrión, y las VMs se ejecutan sobre el hipervisor. **Ejemplos Comunes:** Oracle VirtualBox, VMware Workstation, Parallels Desktop. **Características:** Son más sencillos de instalar y configurar para usuarios individuales, desarrollo local o entornos de prueba. Sin embargo, su rendimiento puede ser ligeramente inferior al Tipo 1 debido a la sobrecarga del sistema operativo anfitrión.

2.5. Componentes Clave en la Gestión de VMs

- **Imágenes ISO:** Archivos de formato estandarizado que contienen una copia exacta de un sistema de archivo. Son ampliamente utilizados como "discos de instalación" virtuales para sistemas operativos dentro de las VMs.
- **Snapshots (Instantáneas):** Representan el estado completo de una máquina virtual (incluyendo memoria, configuración y discos virtuales) en un momento específico. Permiten revertir rápidamente la VM a un estado anterior, lo cual es invaluable para pruebas, desarrollo o recuperación ante errores.
- **Redes Virtuales:** Definen cómo la VM interactúa con el sistema anfitrión, con otras VMs y con la red física externa.
    - **Modo NAT (Network Address Translation):** La VM utiliza la dirección IP del sistema anfitrión para acceder a la red externa (incluyendo internet), pero no es directamente accesible desde fuera de la red del anfitrión. Es el modo predeterminado y el más simple para dar acceso a internet a la VM.
    - **Modo Puente (Bridged Adapter):** La VM se conecta directamente a la red física del anfitrión, obteniendo su propia dirección IP dentro de la misma subred que otros dispositivos. Se comporta como un dispositivo independiente en la red, permitiendo la comunicación bidireccional con otros equipos de la red física.
    - **Red Solo Anfitrión (Host-Only Adapter):** Crea una red privada entre el sistema anfitrión y las VMs. Las VMs pueden comunicarse entre sí y con el anfitrión, pero carecen de acceso directo a la red externa o a internet, siendo útil para entornos de desarrollo aislados.

2.6. Virtualización Tradicional (VMs) vs. Contenedores (Docker)

Aunque ambos buscan encapsular aplicaciones y sus dependencias, operan a niveles diferentes de la pila de software, lo que les confiere ventajas y desventajas específicas:

Aspecto	Virtualización (VirtualBox/VMware)	Contenedores (Docker)
Arquitectura	Cada VM incluye un SO completo	Comparten el kernel del SO anfitrión
Nivel de Abstracción	Hardware (emulado o virtualizado)	Sistema Operativo
Consumo de Recursos	Mayor consumo de CPU, RAM, y espacio en disco por cada SO invitado completo	Mínimo *overhead*, muy ligeros y eficientes en el uso de recursos
Aislamiento	Alto (cada VM es un sistema independiente)	Aislamiento a nivel de proceso (no de kernel)
Tiempo de Inicio	Lento (requiere el arranque de un SO completo, minutos)	Instantáneo (los contenedores se inician en segundos)

2.7. Beneficios y Desafíos de la Virtualización

Beneficios:

- **Consolidación de Hardware y Ahorro de Costos:** Permite ejecutar múltiples sistemas en un único servidor físico, reduciendo la necesidad de comprar y mantener hardware adicional.
- **Flexibilidad y Escalabilidad:** Facilita la rápida creación, clonación, migración y eliminación de VMs, lo que permite adaptar la infraestructura a las demandas cambiantes.
- **Recuperación ante Desastres (DR) y Alta Disponibilidad (HA):** Los *snapshots*, la clonación y la migración de VMs simplifican la recuperación de sistemas ante fallos y la implementación de soluciones de alta disponibilidad.
- **Aislamiento y Seguridad:** Cada VM opera en un entorno aislado, lo que previene que fallos o vulnerabilidades en una VM afecten a otras o al sistema anfitrión.
- **Desarrollo y Pruebas:** Proporciona entornos de desarrollo y pruebas consistentes y aislados, sin impactar el sistema operativo principal del desarrollador.

Desafíos:

- **Complejidad de Gestión:** La administración de un gran número de VMs y sus recursos asociados puede ser compleja y requerir herramientas de orquestación especializadas.
- **Sobrecarga de Rendimiento:** Aunque los hipervisores son eficientes, la capa de virtualización puede introducir una ligera latencia o consumo adicional de recursos que podría afectar a aplicaciones extremadamente sensibles al rendimiento.
- **Puntos Únicos de Fallo (SPOF):** Un fallo crítico en el hipervisor o en el hardware físico subyacente puede impactar a todas las VMs que aloja.
- **Licenciamiento:** La virtualización de algunos sistemas operativos o software puede implicar consideraciones de licenciamiento adicionales.

---

3. Caso Práctico: Configuración de Servidor Web con Ubuntu Desktop en VirtualBox

El objetivo principal de este caso práctico fue la instalación de una máquina virtual (VM) con **Ubuntu Desktop 24.04.2 LTS** y la posterior configuración de un servidor web Apache2 en dicho entorno.

3.1. Procedimiento de Creación de la Máquina Virtual

1. **Inicio del Asistente:** Se inició el asistente de "Nueva Máquina Virtual" en la interfaz gráfica de VirtualBox.
2. **Configuración Básica:**
    - **Nombre:** Se asignó el nombre "UBUNTU-AYSO" a la máquina virtual para una fácil identificación.

    - **Carpeta:** Se especificó la ubicación predeterminada para el almacenamiento de archivos de la VM: `C:\Users\u7ser\.VirtualBox VMs`.
    - **Imagen ISO:** Se seleccionó la imagen `ubuntu-24.04.2-desktop-amd64.iso` previamente descargada.
    - **Tipo y Versión de SO:** VirtualBox detectó automáticamente el tipo de SO como "Linux" y la versión como "Ubuntu (64-bit)" basado en la imagen ISO proporcionada.
3. **Asignación de Recursos de Hardware:**
    - **Memoria RAM:** Se asignaron **3072 MB (3 GB) de memoria RAM** a la VM. Esta cantidad se considera adecuada para una instalación de Ubuntu Desktop 24.04.2 LTS y la ejecución de un servidor Apache básico sin comprometer excesivamente el rendimiento del sistema anfitrión (Windows 10/11 en este caso).
    - **Procesadores:** Se dejó la configuración predeterminada de 1 CPU virtual (o se ajustó a 2 si el host lo permitía holgadamente), lo cual es suficiente para esta carga de trabajo.
4. **Creación del Disco Duro Virtual:**
    - Se optó por crear un nuevo disco duro virtual.
    - **Tipo de Archivo:** Se seleccionó **VDI (VirtualBox Disk Image)**, el formato nativo de VirtualBox.
    - **Almacenamiento:** Se eligió **"Tamaño fijo"** para garantizar un rendimiento ligeramente superior y evitar el crecimiento dinámico del archivo de disco.
    - **Tamaño:** Se asignaron **50 GB** de espacio en disco. Esta capacidad es más que suficiente para una instalación de Ubuntu Desktop y los archivos de Apache, dejando espacio para logs y futuros servicios básicos.

3.2. Instalación del Sistema Operativo (Ubuntu Desktop 24.04.2 LTS)

1. **Montaje de ISO:** Una vez creada la VM, se verificó que la imagen ISO de **Ubuntu Desktop 24.04.2 LTS** estuviera montada como unidad óptica en la configuración de la VM.
2. **Inicio de la VM:** Se inició la máquina virtual para comenzar el proceso de instalación de Ubuntu Desktop.
3. **Configuración de Red Inicial:** Durante la instalación, se permitió que el instalador configurara la red de forma automática mediante DHCP, lo que facilitó la obtención de una dirección IP en el modo red configurado previamente.
4. **Creación de Usuario:** Se creó un usuario estándar (ej. `joana`) con permisos `sudo` para las tareas administrativas.

3.3. Configuración del Servidor Web Apache2

Una vez finalizada la instalación de **Ubuntu Desktop 24.04.2 LTS** y tras el reinicio, se procedió con la configuración de Apache desde la terminal de la VM:

Actualización de Repositorios y Paquetes: Para asegurar que se instalaran las versiones más recientes de los paquetes y sus dependencias, se ejecutaron los siguientes comandos:

Bash
```
sudo apt update        # Actualiza la lista de paquetes disponibles desde los repositorios.
sudo apt upgrade -y    # Actualiza todos los paquetes instalados a sus últimas versiones.
```

1.

Instalación de Apache2: Se instaló el paquete del servidor web Apache:
Bash
```
sudo apt install apache2 -y # Instala el servidor web Apache2. El '-y' confirma automáticamente la instalación.
```

2.

Habilitación del Firewall (UFW): Ubuntu Desktop 24.04.2 LTS viene con el firewall UFW (Uncomplicated Firewall) preinstalado. Se configuró para permitir el tráfico HTTP y HTTPS, esencial para un servidor web:
Bash
```
sudo ufw allow 'Apache Full' # Abre los puertos 80 (HTTP) y 443 (HTTPS) para el perfil 'Apache Full'.
sudo ufw enable              # Habilita el firewall (si no estaba ya habilitado).
sudo ufw status verbose      # Verifica el estado del firewall y las reglas activas.
```

3. **Observación:** El perfil 'Apache Full' es conveniente ya que abre ambos puertos necesarios para un servidor web.

---

4. Metodología Utilizada

La implementación de este trabajo práctico siguió una metodología basada en los siguientes pilares:

- **Preparación del Entorno Host:** Se realizó la instalación de **Oracle VirtualBox 7.1.8** (o versión similar) en un sistema operativo anfitrión **Windows**. La descarga se obtuvo directamente desde el sitio web oficial de VirtualBox.
- **Obtención de Recursos del Invitado:** Se procedió a la descarga de la imagen ISO oficial de **Ubuntu Desktop 24.04.2 LTS**. Se eligió la versión LTS (Long Term Support) por su estabilidad y el soporte extendido.
- **Validación y Pruebas:** Tras la configuración de Apache, se realizó una verificación exhaustiva del servicio web. Esto incluyó:
    - Obtención de la dirección IP de la VM (ej. con el comando `ip a` en la terminal de Ubuntu Desktop).
    - Acceso a dicha dirección IP desde un navegador web en el sistema anfitrión. Se esperaba y se obtuvo la página de bienvenida predeterminada de Apache ("It works!"), confirmando el correcto funcionamiento del servidor web y la conectividad de red.

---

5. Dificultades Encontradas y Soluciones

Durante el desarrollo del proyecto, se enfrentaron algunas dificultades operativas y de configuración:

- **Requisito de Microsoft Visual C++ 2019:**
    - **Problema:** Al iniciar el proceso de instalación de VirtualBox, se presentó un mensaje que indicaba la necesidad de tener instalada la redistribuible de Microsoft Visual C++ 2019. Sin esta dependencia, la instalación de VirtualBox no podía continuar.
    - **Resolución:** Para superar esta dificultad, se procedió a **descargar e instalar** la redistribuible de Microsoft Visual C++ 2019 desde el sitio web oficial de Microsoft. Una vez instalada esta dependencia, el proceso de instalación de VirtualBox pudo reanudarse y completarse de manera exitosa.
- **Problemas de Distribución de Teclado en la Terminal de la VM:**
    - **Problema:** Al intentar escribir comandos y código Python directamente desde la terminal de **Ubuntu Desktop 24.04.2 LTS** dentro de la máquina virtual, se detectó una discrepancia en la distribución de las teclas. Por ejemplo, el signo de igual (`=`) se encontraba mapeado a la tecla del signo de interrogación (`?`) en el teclado físico, y los dos puntos (`:`) se ubicaban en la tecla de la letra `ñ`. Esto dificultó la escritura fluida de comandos y el desarrollo de scripts.
    - **Impacto:** Aunque inicialmente frustrante y ralentizador, la dificultad se superó mediante la memorización de las ubicaciones corregidas de los caracteres necesarios. No se implementó una solución permanente de remapeo de teclado dentro de la VM, pero el código pudo completarse exitosamente a pesar de la anomalía. Para futuras implementaciones, se podría considerar la configuración del *layout* de teclado directamente en Ubuntu Desktop 24.04.2 LTS (por ejemplo, con `sudo dpkg-reconfigure keyboard-configuration` o `setxkbmap latam`) o en la configuración de entrada de VirtualBox.

---

6. Resultados Obtenidos

Los resultados del caso práctico demostraron el éxito de la implementación:

- **Ejecución Correcta del SO Invitado:** El sistema operativo **Ubuntu Desktop 24.04.2 LTS** se instaló y ejecutó de manera estable y funcional dentro del entorno de

VirtualBox. Se pudo acceder a la terminal de la VM, iniciar sesión con el usuario creado y ejecutar comandos sin problemas.

- **Servidor Apache Disponible en Red Local:** El servidor web Apache2 fue configurado correctamente y se encontraba completamente operativo. Se confirmó su accesibilidad desde el sistema anfitrión (Windows) y otros dispositivos en la misma red local mediante la dirección IP asignada a la VM. La página de bienvenida predeterminada de Apache se cargó sin inconvenientes.
- **Comprensión de Interacción Host-Invitado:** La experiencia práctica permitió a los alumnos comprender en detalle la interacción y conectividad entre el sistema operativo anfitrión (host) y el sistema operativo invitado (VM), así como el rol del hipervisor en la gestión de recursos y la comunicación de red.

---

7. Conclusiones

La virtualización, facilitada por herramientas accesibles y potentes como Oracle VirtualBox, se ha revelado como una tecnología indispensable para la creación de entornos de prueba, desarrollo y aprendizaje, eliminando la necesidad de invertir en hardware adicional. La experiencia práctica con **Ubuntu Desktop 24.04.2 LTS** y Apache ha permitido a los alumnos consolidar conocimientos teóricos y desarrollar habilidades técnicas en la configuración y gestión de entornos virtualizados. A pesar de enfrentar desafíos menores como la distribución del teclado y dependencias de software, la capacidad de resolver estos problemas de manera autónoma ha reforzado el aprendizaje. Este proyecto sienta una base sólida para la comprensión y aprovechamiento de la virtualización en futuros entornos profesionales y académicos.

---

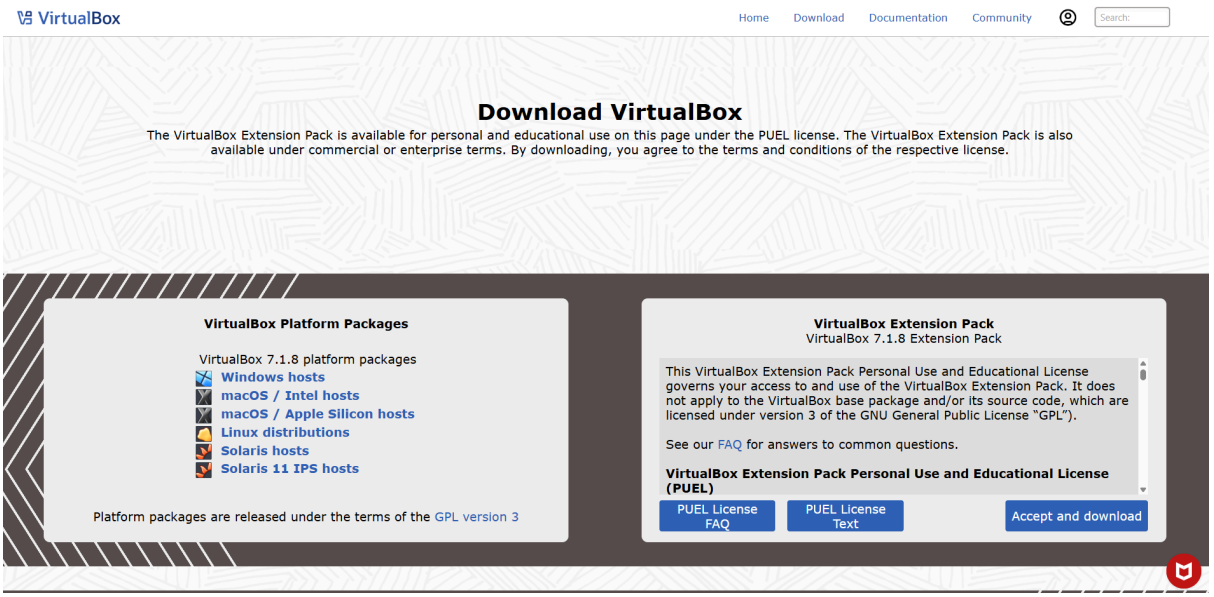
8. Bibliografía

- Documentación oficial de VirtualBox: https://www.virtualbox.org/manual/
- Guía oficial de Ubuntu Server: https://ubuntu.com/server/docs
- Redes virtuales con VirtualBox (ArchWiki): https://wiki.archlinux.org/title/VirtualBox

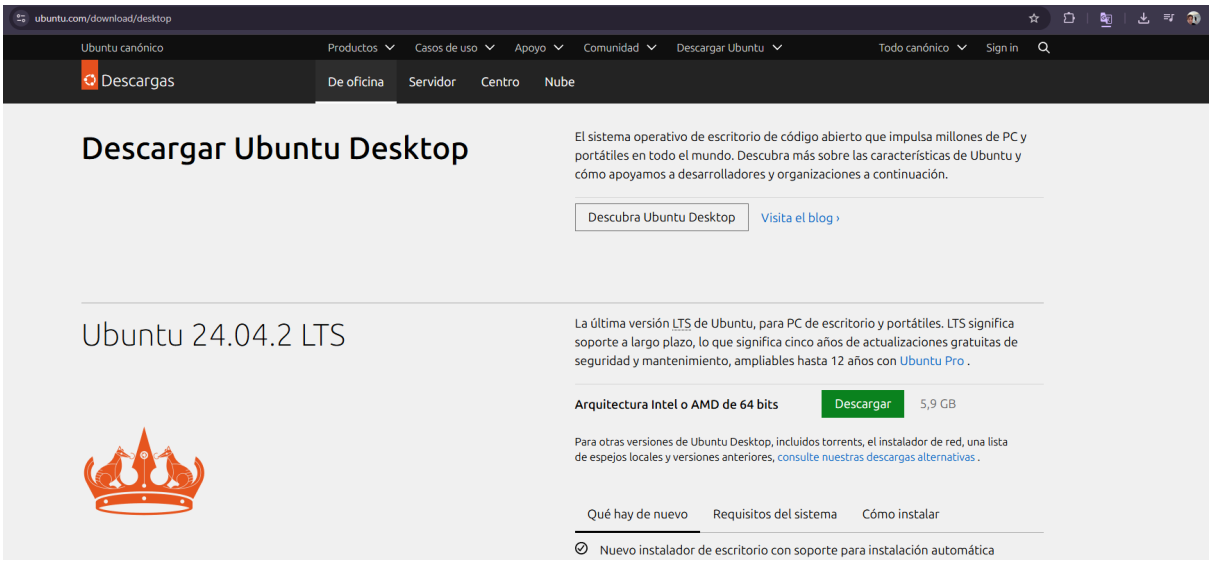
---

9. Anexos

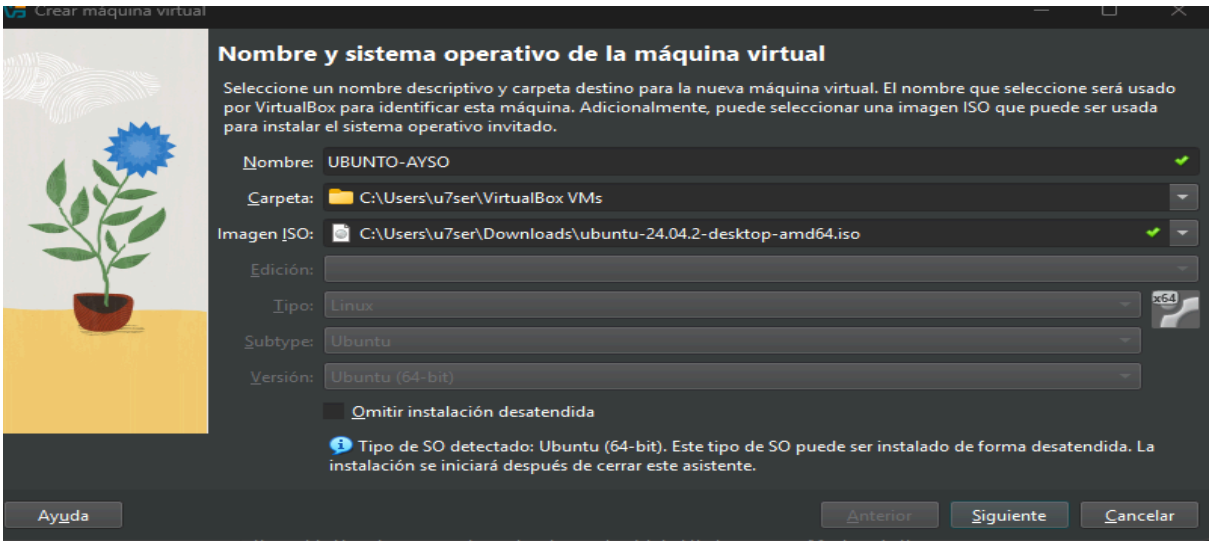
En esta sección se incluirán las capturas de pantalla que evidencian los pasos y resultados del proyecto. Se proporcionarán pies de foto descriptivos para cada imagen.



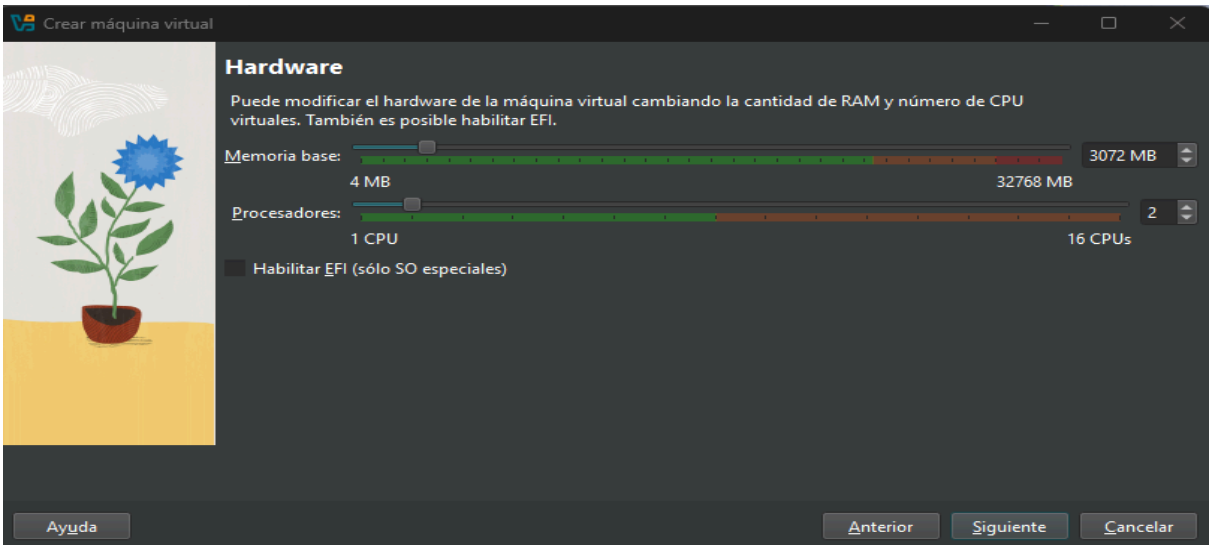
- [Imagen: Página de descarga de VirtualBox]



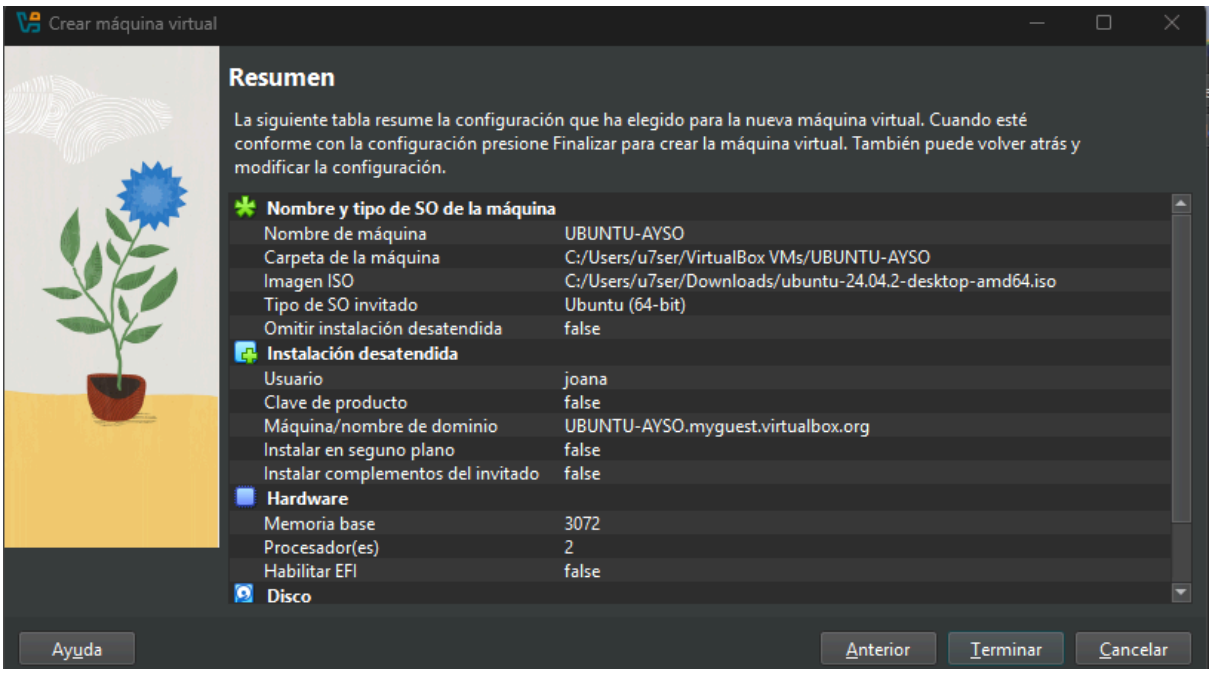
- [Imagen: Página de descarga de ubuntu]

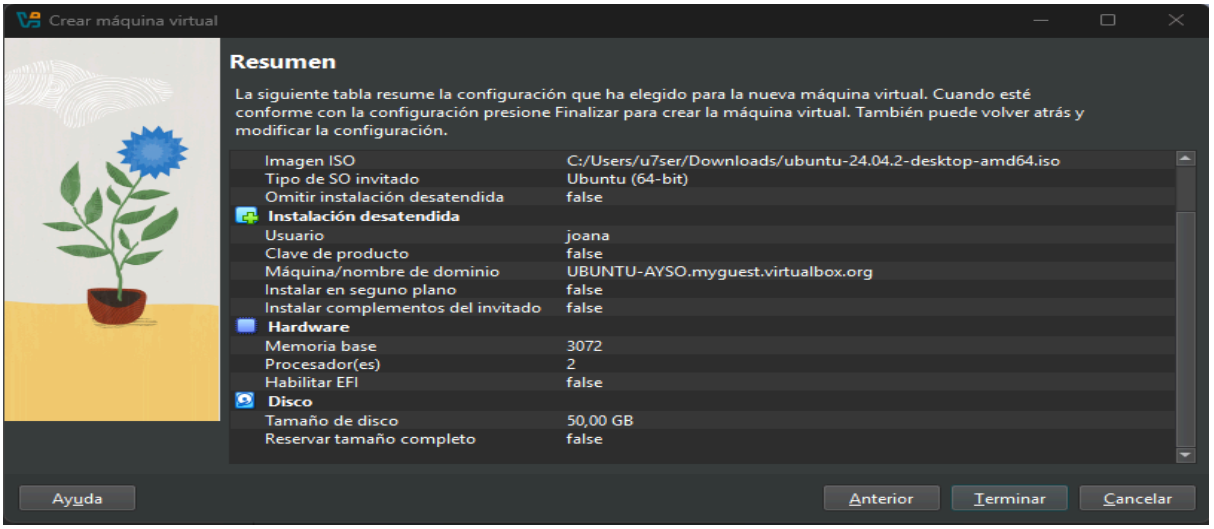
- 

  - [Imagen: Ventana "Crear máquina virtual" - Nombre y sistema operativo, con la ISO seleccionada]

- 

  - [Imagen: Ventana "Crear máquina virtual" - Asignación de memoria RAM]

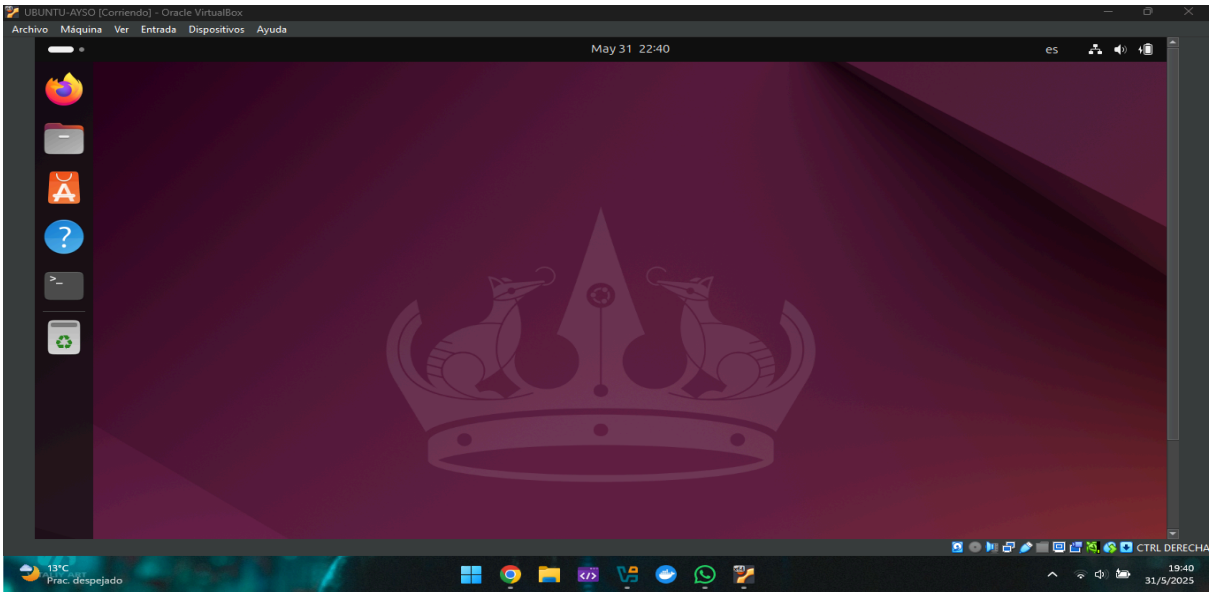
- 

- 

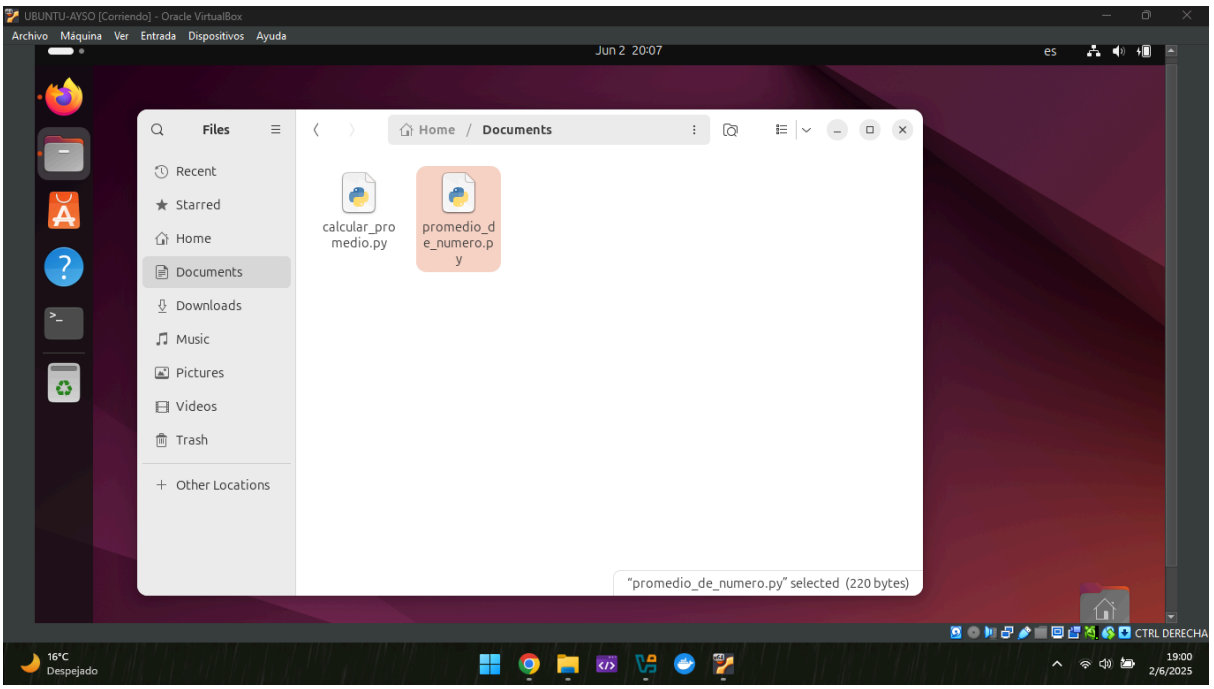
- [Imagen: Configuración de la VM en VirtualBox - Pestaña "Sistema" (general)]

- 

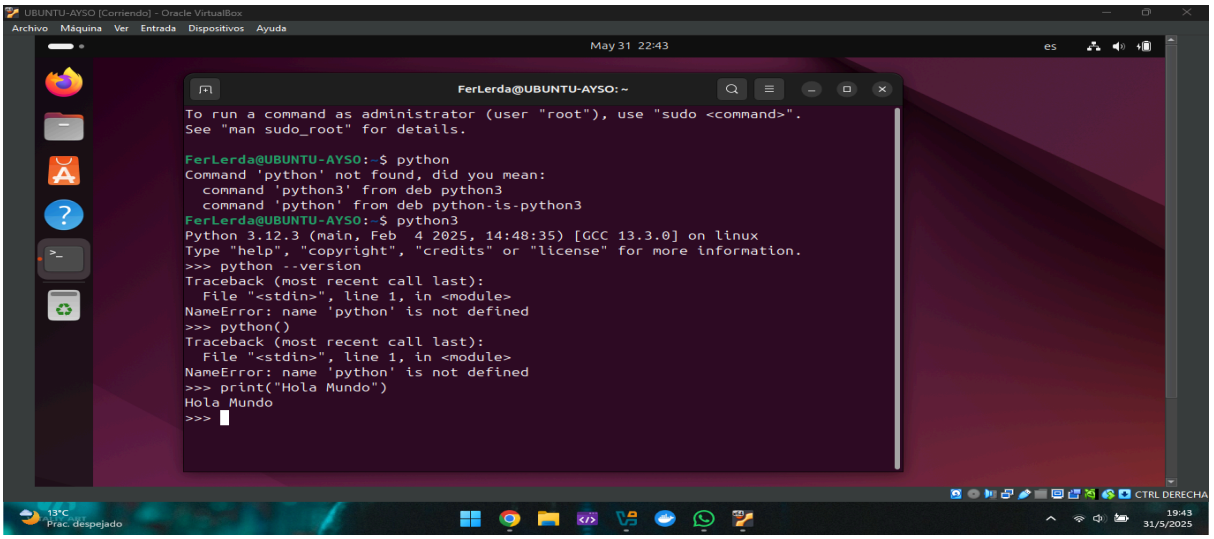
  - [Imagen: Ventana de inicio de Ubuntu]

- 

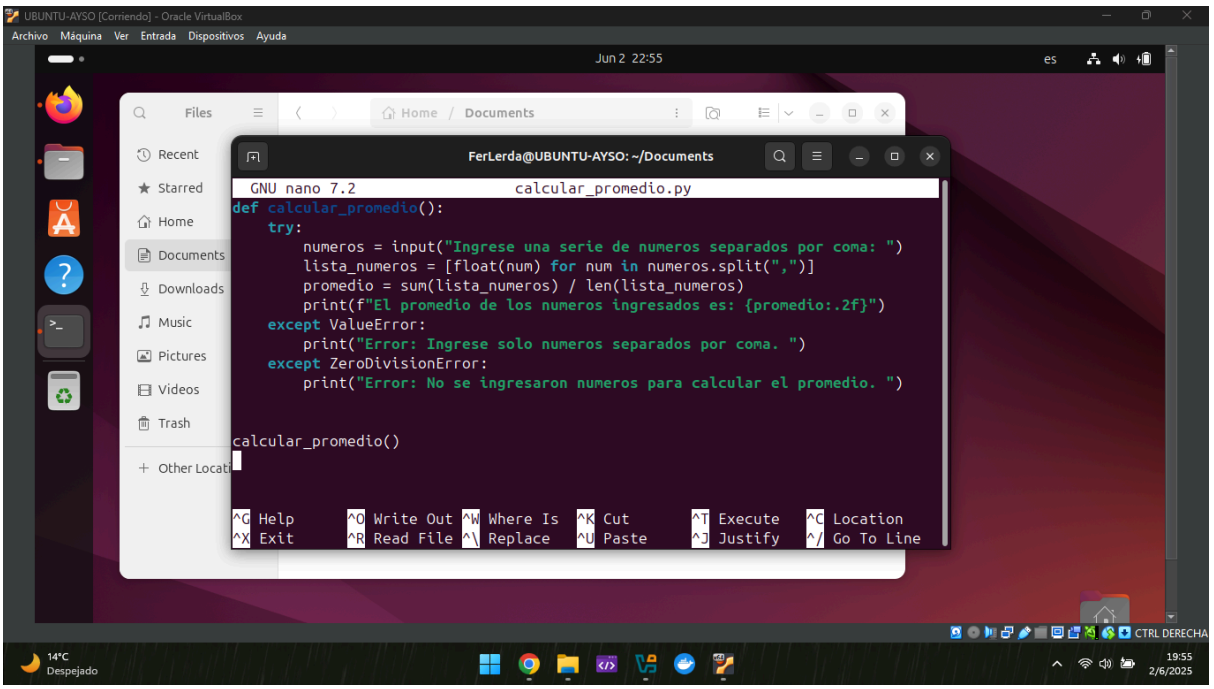
  - [Imagen: Escritorio de ubuntu]

- 

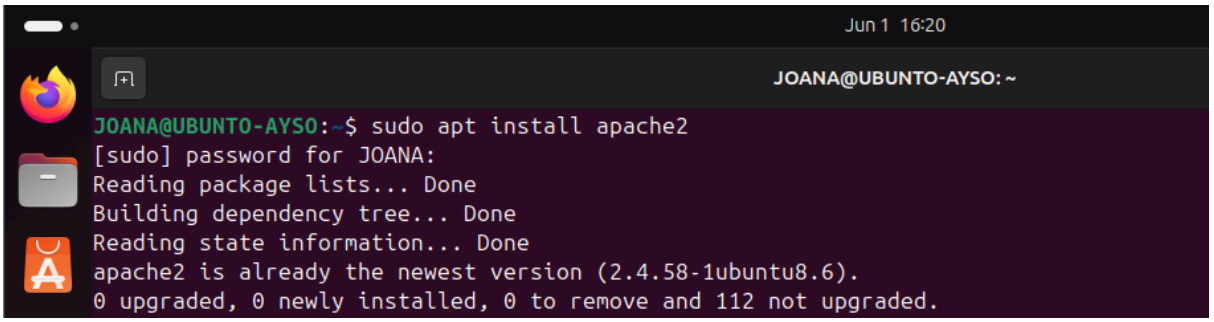
  - [Imagen: Creación del programa promedio_de_numeros.py]

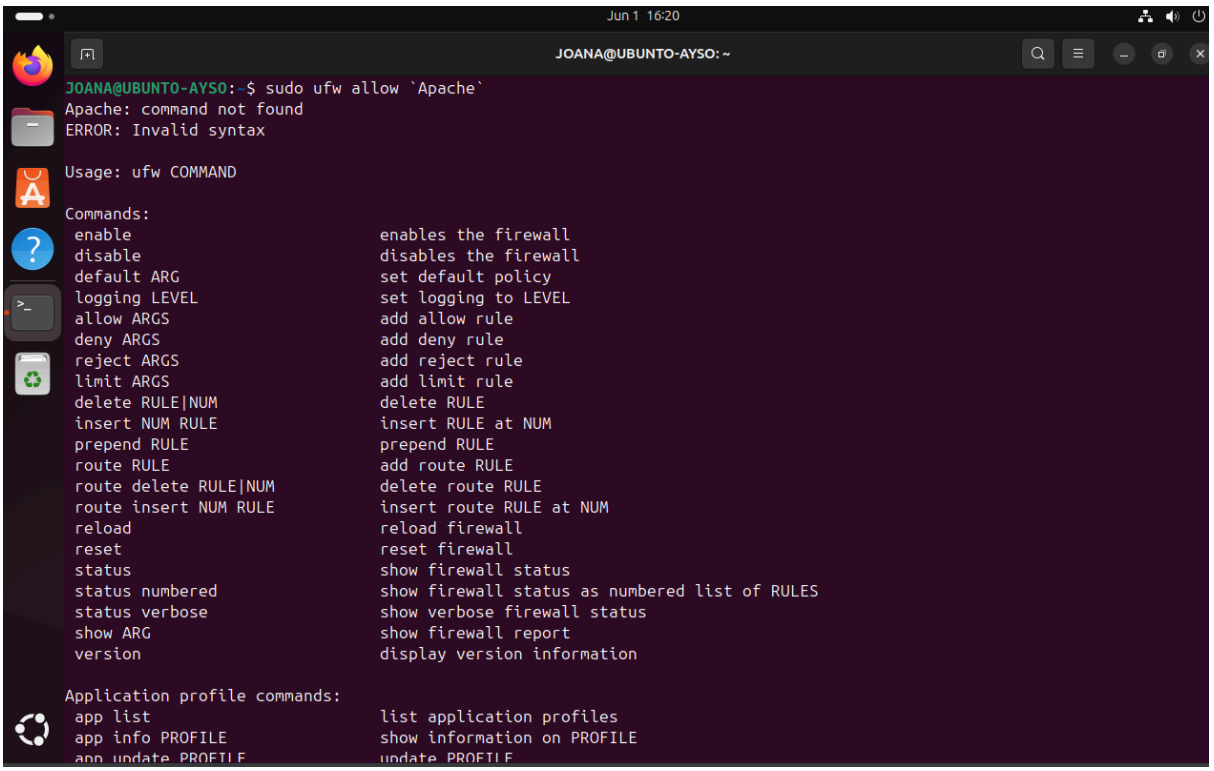
- 

  - [Imagen: Terminal de ubuntu]

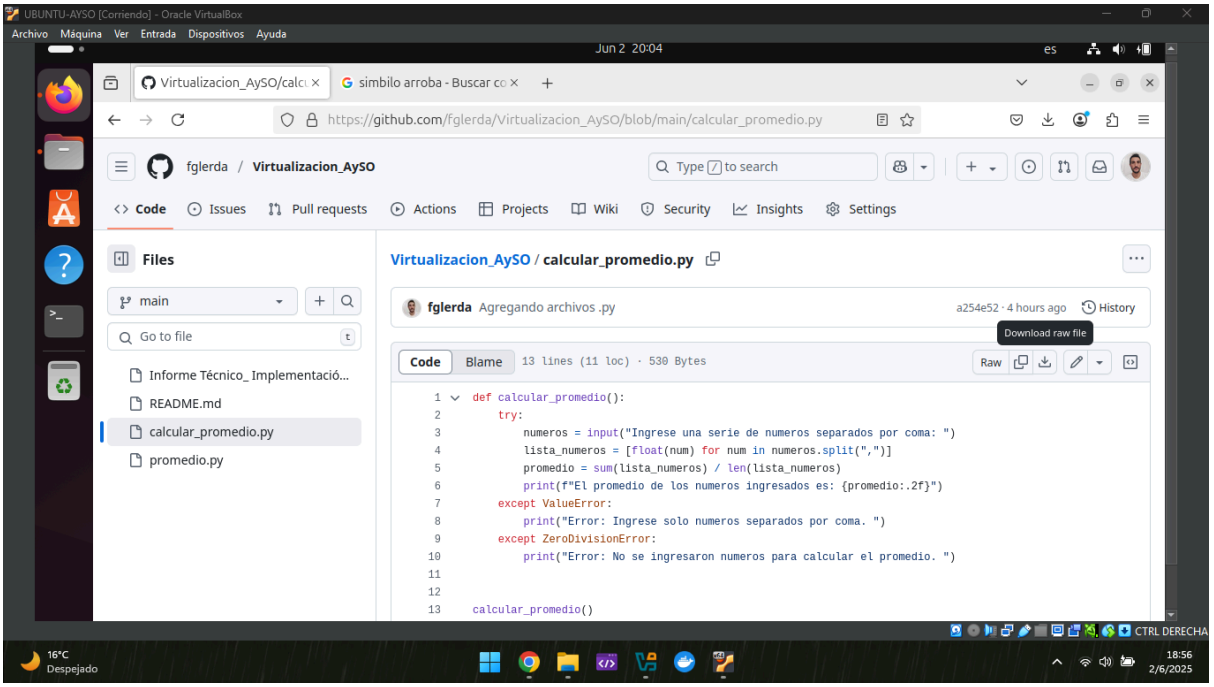
- 

  - [Imagen: Ventana GNU nano 7.2]

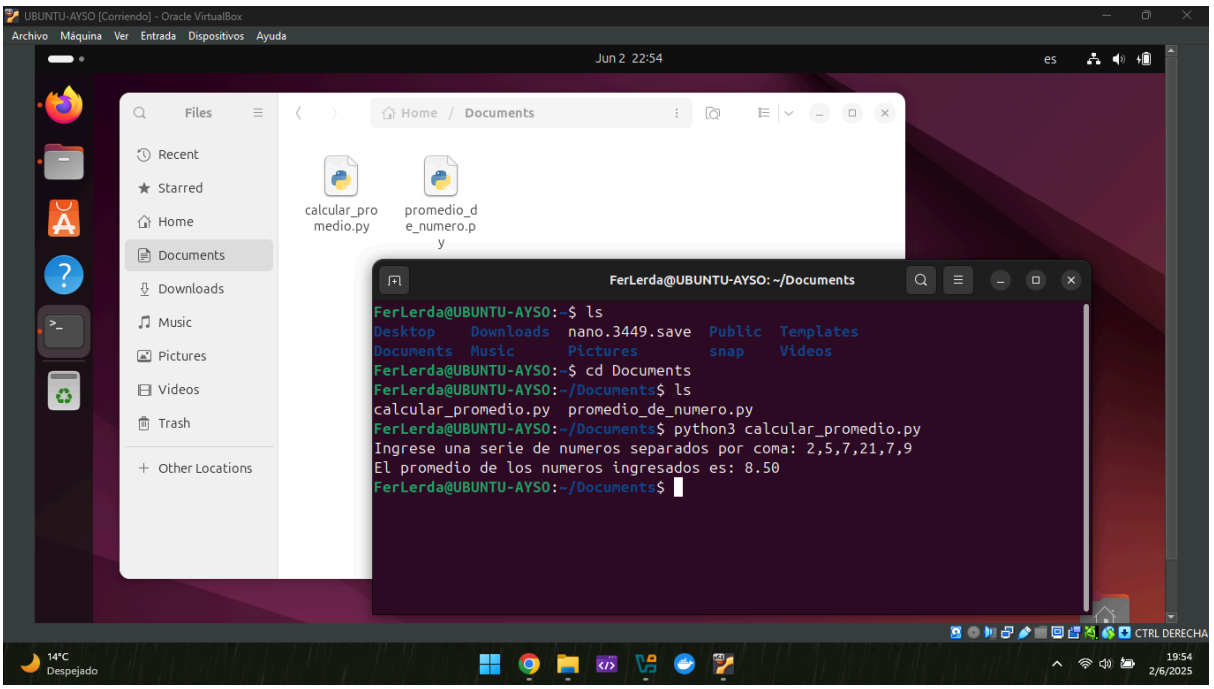
- 

- 

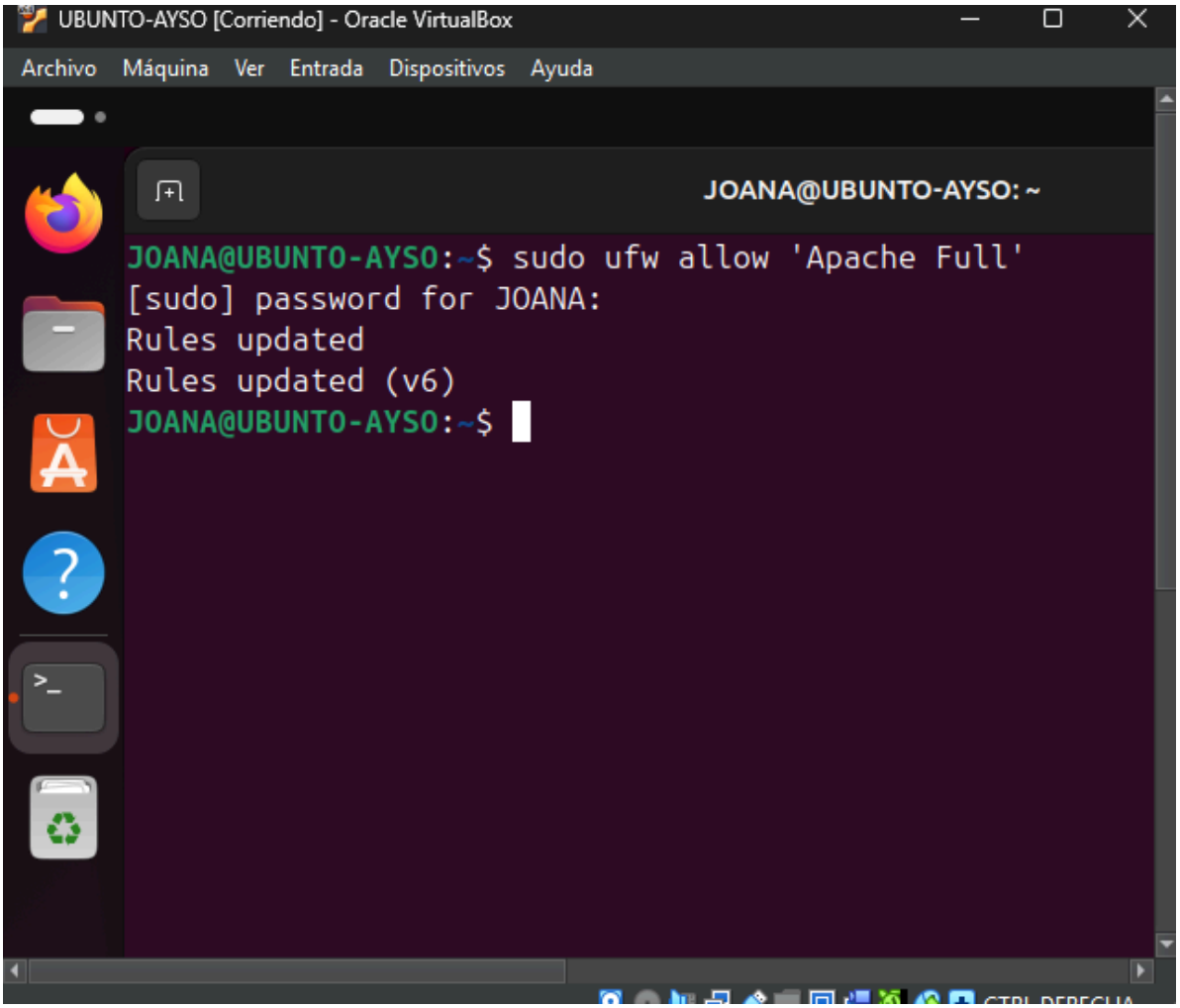
  - [Imagen: Configuración Apache]

- 

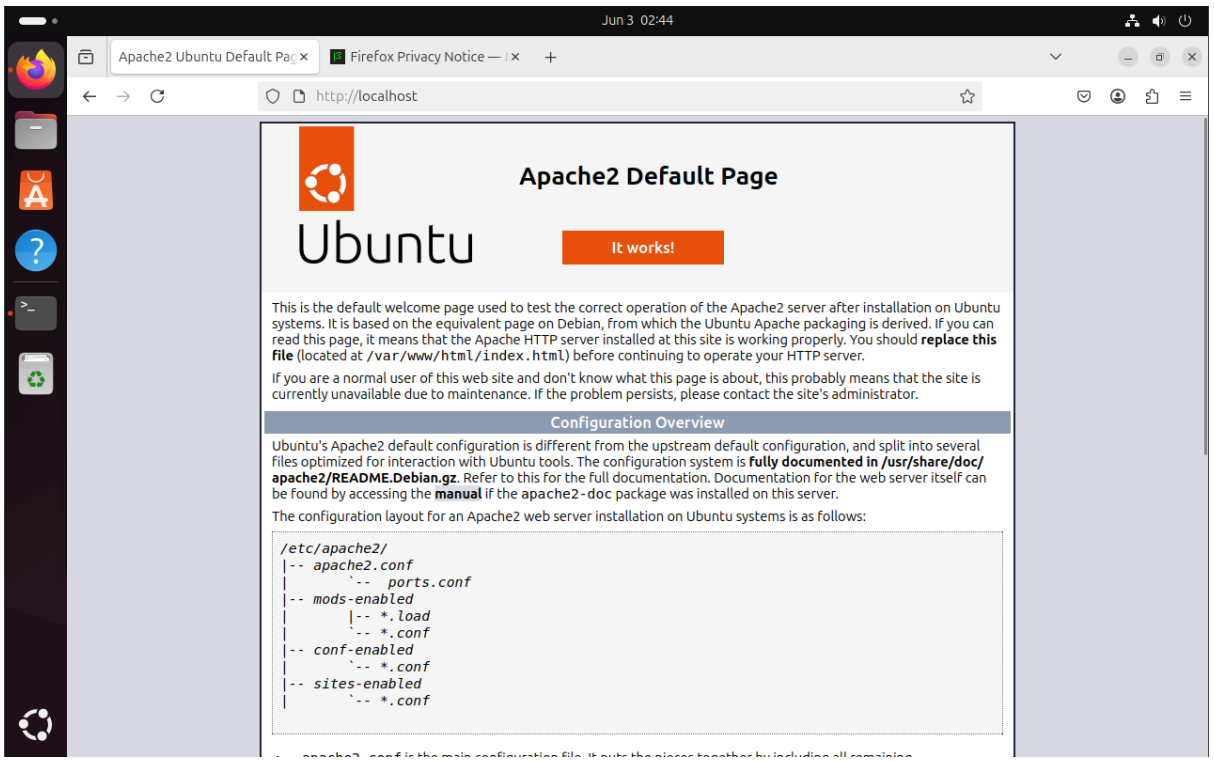
  - [Imagen: Repositorio de GitHub desde navegador mozilla en Ubuntu]



- [Imagen: Ejecucion de calcular_promedio.py desde la consola bash en Ubuntu]



- [Imagen: ejecución de Apache]



- [Imagen: página web por defecto de servidor Apache2]