



**Universidad de Guadalajara**  
**Centro Universitario de Ciencias Exactas E Ingenierías**  
**Ingeniería Informática**

**Juan Antonio Ramírez Aguilar**  
**(212482507)**

**Seminario de Solución de Problemas de Uso, Adaptación,  
Explotación de Sistemas Operativos**

**Sección: D02**

**Mtra. Becerra Velázquez Violeta del Rocío**

**“Actividad 5 (1.5 Concepto de máquina virtual y partición  
de disco)”**

## Indicé

<b>Máquinas Virtuales</b> .....	4
<b>¿Cuáles son sus principales aplicaciones?</b> .....	5
<b>¿Cuáles hay?</b> .....	6
Hipervisores de tipo 1 .....	6
Hipervisores de tipo 2 .....	6
<b>¿Existe más de un tipo de Máquina Virtual?</b> .....	7
Máquinas virtuales VMware .....	7
Máquinas virtuales Windows .....	7
Máquinas virtuales Android .....	7
Máquinas virtuales Mac .....	7
Máquinas virtuales iOS .....	8
Máquinas virtuales Java .....	8
Máquinas virtuales Python .....	8
Máquinas virtuales Linux .....	9
Máquinas virtuales Ubuntu .....	9
<b>¿Qué es un hipervisor o Virtual Machine Monitor (VMM)?</b> .....	9
<b>Hipervisores más utilizados</b> .....	9
<b>Instalación de Hipervisor</b> .....	10
<b>Instalación de sistemas operativos</b> .....	11
<b>Sistema operativo de Fabrica</b> .....	11
Instalación de Windows 11 .....	11
<b>Sistema operativo en Dual Boot</b> .....	11
<b>Sistema operativo en máquina virtual</b> .....	13
Instalación de Debian en Máquina Virtual .....	13
<b>Video de demostración</b> .....	17
<b>Conclusiones</b> .....	17
<b>Referencias</b> .....	17

## Contenido de imágenes

Imagen 1. Descarga de VirtualBox .....	10
Imagen 2. Página de descarga de Windows 11.....	11
Imagen 3. Versiones de Pop!_OS .....	12
Imagen 4. Particiones de mi disco duro .....	12
Imagen 5. Página web de Debian .....	13
Imagen 6. Configuración de Debian .....	13
Imagen 7. Máquina virtual creada .....	14
Imagen 8. Configuración de Instalación Debian.....	14
Imagen 9. Uso del disco en Debian .....	15
Imagen 10. Arranque de Debian .....	16
Imagen 11. Sistema operativo Debian .....	16

## Máquinas Virtuales

Una máquina virtual (virtual machine) es una representación virtual o emulación de un equipo físico que emplea software en lugar de hardware para ejecutar programas e implementar aplicaciones.

Las máquinas virtuales funcionan mediante virtualización, un proceso de creación de versiones virtuales o basadas en software de recursos (cálculo, almacenamiento, redes, servidores) o aplicaciones.

La virtualización permite un uso más eficiente del hardware informático físico y es fundamental para la computación en la nube.

La virtualización es posible con un hipervisor, también conocido como monitor de máquina virtual (VMM). Esta ligera capa de software administra las máquinas virtuales a medida que se ejecutan entre sí.

El nacimiento de la virtualización se remonta a 1964, cuando IBM diseñó y presentó CP-40, un proyecto experimental de investigación de tiempo compartido para IBM System/360. El CP-40, que luego se convirtió en el CP-67 y luego en Unix, proporcionó hardware informático capaz de admitir múltiples usuarios simultáneos y sentó las bases para las máquinas virtuales.

El 2 de agosto de 1972, IBM implementó lo que muchos consideran como la primera máquina virtual, la VM/370, y los primeros mainframes System/370 que admitieron memoria virtual.

En 1998, VMware desarrolló el sistema operativo x86, que permitió segmentar una sola máquina en varias máquinas virtuales, cada una con su propio sistema operativo. En 1999, la empresa lanzó VM Workstation 1.0, el primer producto comercial que permitía a los usuarios ejecutar múltiples sistemas operativos como máquinas virtuales en una sola PC.

Hoy en día, la virtualización es una práctica estándar para la infraestructura de TI de nivel empresarial y una fuerza impulsora en la economía de la computación en la nube, lo que permite a las empresas impulsar una mayor utilización de la capacidad y reducir los costos. Toda la infraestructura de TI se puede virtualizar, incluidos los entornos de escritorio, los sistemas operativos, el hardware de almacenamiento, los centros de datos, entre otros.

## ¿Cuáles son sus principales aplicaciones?

Las VM tienen una amplia gama de usos tanto para los administradores de TI de las empresas como para los usuarios, entre los que se incluyen los siguientes:

- **Habilitar la computación basada en la nube:** las VM son la unidad fundamental de la computación en la nube, ya que permiten que docenas de aplicaciones y cargas de trabajo se ejecuten y escalen con éxito.
- **Acelerar la migración de cargas de trabajo:** debido a su portabilidad, las VM ayudan a acelerar la migración de cargas de trabajo desde entornos on premises hasta entornos basados en la nube.
- **Acelerar los recorridos a la nube híbrida:** las VM proporcionan la infraestructura para crear entornos de nube híbrida que combinan entornos on premises, de nube privada y nube pública en una única infraestructura flexible de TI.
- **Compatibilidad con DevOps:** las VM son una excelente manera de respaldar a los equipos de DevOps y a otros desarrolladores empresariales, ya que les permiten configurar plantillas de VM con la configuración para sus procesos de desarrollo y pruebas de software. Pueden crear VM para tareas, como pruebas de software estáticas, incluidos estos pasos en un flujo de trabajo de desarrollo automatizado. Estas funciones ayudan a optimizar la cadena de herramientas de DevOps.
- **Probar un nuevo sistema operativo:** una VM permite probar un nuevo sistema en su escritorio sin afectar su SO principal.
- **Investigar malware:** las VM son útiles para los investigadores de malware que, con frecuencia, necesitan máquinas nuevas para probar programas maliciosos.
- **Ejecutar software incompatible:** algunos usuarios deben usar un SO y seguir necesitando un programa que solo está disponible en otro.
- **Navegar de forma segura:** el uso de una VM para navegar permite visitar sitios sin preocuparse por las infecciones. Puede tomar una instantánea de su máquina y volver a ella luego de cada sesión de navegación. Los usuarios podrían configurar este escenario de navegación mediante un hipervisor de escritorio tipo 2. Alternativamente, un administrador podría proporcionar un escritorio virtual temporal en el servidor.
- **Admitir la recuperación ante desastres (DR):** con un entorno virtualizado, es fácil aprovisionar y desplegar recursos, lo que le permite replicar o clonar la VM cuando sea necesario. Este proceso ocurre en minutos, a diferencia de las muchas horas que se necesitan para aprovisionar y configurar un nuevo servidor físico, lo cual es crucial para la DR.

## ¿Cuáles hay?

Existen dos tipos principales de hipervisores:

### Hipervisores de tipo 1

Se ejecutan directamente en el hardware físico (en general, un servidor), que reemplaza el SO. Normalmente, se emplea un producto de software independiente para crear y manipular VM en el hipervisor.

Algunas herramientas de gestión, como **vSphere** de VMware, le permiten seleccionar un sistema operativo invitado para instalarlo en la VM. Puede usar una VM como plantilla para otras y duplicarla para crear otras nuevas. Según sus necesidades, puede crear varias plantillas de VM para diferentes propósitos, como pruebas de software, bases de datos de producción o entornos de desarrollo. Una máquina virtual basada en **kernel (KV)** es un ejemplo de hipervisor de tipo 1.

### Hipervisores de tipo 2

Se ejecutan como una aplicación dentro de un sistema operativo host y, por lo general, se dirigen a plataformas de escritorio o portátiles de un solo usuario.

Con un hipervisor de tipo 2, se crea manualmente una VM y se instala un SO invitado en su interior. Puede usar el hipervisor para asignar recursos físicos a su VM, lo cual configura manualmente la cantidad de núcleos de procesador y memoria que puede usar. Según las capacidades del hipervisor, puede establecer opciones, como el aceleramiento 3D para gráficos. Los hipervisores de tipo 2 incluyen **VMware Workstation** y **Oracle VirtualBox**.

## ¿Existe más de un tipo de Máquina Virtual?

Existen varios tipos de máquinas virtuales. Muchos de ellos se enfocan en casos específicos de uso.

### **Máquinas virtuales VMware**

VMware, la primera empresa en comercializar con éxito la virtualización de la arquitectura de microprocesador x86, es líder en el mercado de la virtualización. VMware proporciona hipervisores tipo 1 y tipo 2, y software de VM a clientes empresariales.

### **Máquinas virtuales Windows**

La mayoría de los hipervisores admiten VM que ejecutan el SO Windows como invitado. El hipervisor Hyper-V de Microsoft viene como parte del SO Windows. Cuando se instala, crea una partición principal que se contiene a sí misma y al SO Windows principal, cada uno de los cuales obtiene acceso privilegiado al hardware. Otros SO, incluidos los invitados de Windows, se ejecutan en particiones secundarias y se comunican con el hardware a través de la partición principal.

### **Máquinas virtuales Android**

El SO Android de código abierto de Google es común en dispositivos domésticos móviles y conectados.

El SO Android solo se ejecuta en la arquitectura de procesador ARM típica de estos dispositivos, pero es posible que los entusiastas, los jugadores de Android o los desarrolladores de software prefieran ejecutarlo en PC.

Esta situación puede ser problemática porque las PC se ejecutan en una arquitectura de procesador x86 completamente diferente, y un hipervisor de virtualización de hardware solo pasa instrucciones entre la máquina virtual y la CPU. No los traduce para procesadores con diferentes conjuntos de instrucciones.

Varios proyectos, como Shashlik o Genymotion, pueden abordar este problema mediante el uso de un emulador que recrea la arquitectura ARM en el software. Una alternativa, el proyecto Android-x86, traslada Android a la arquitectura x86. Para ejecutarlo, debe instalarse el programa Android-x86 como una máquina virtual que usa el hipervisor VirtualBox tipo 2. Otra alternativa, Anbox, ejecuta el SO Android en el kernel de un SO Linux host.

### **Máquinas virtuales Mac**

Apple permite que su sistema macOS solo se ejecute en hardware de Apple. Esto significa que no puede ejecutarse en hardware que no sea de Apple como una VM o bajo su acuerdo de licencia de

usuario final. Sin embargo, puede usar hipervisores tipo 2 en hardware Mac para VM con un invitado macOS.

## **Máquinas virtuales iOS**

Hoy en día, es imposible ejecutar iOS en una VM porque Apple controla estrictamente su SO iOS y solo permite que se ejecute en dispositivos iOS.

Lo más parecido a una máquina virtual iOS es el simulador de iPhone que se envía con el entorno de desarrollo integrado Xcode, que simula todo el sistema iPhone en software.

## **Máquinas virtuales Java**

La plataforma Java es un entorno de ejecución para programas escritos en el lenguaje de desarrollo de software Java. La promesa de Java (“escribir una vez, ejecutar en cualquier lugar”) significa que cualquier programa Java podría ejecutarse en cualquier plataforma Java, razón por la cual la plataforma Java incluía una máquina virtual Java (JVM).

Los programas Java contienen bytecode, que es una forma de instrucción destinada a la JVM. La JVM compila este bytecode en código de máquina, que es el lenguaje de nivel más bajo que usa la computadora host. La JVM en la plataforma Java de una plataforma informática crea un conjunto diferente de instrucciones de código de máquina que la JVM de otra, en función del código de máquina que espera el procesador.

Por lo tanto, la JVM no ejecuta un SO completo ni usa un hipervisor como lo hacen otras VM. En cambio, traduce los programas de software a nivel de aplicación para que se ejecuten en un hardware particular.

## **Máquinas virtuales Python**

Al igual que la JVM, la máquina virtual Python no se ejecuta en un hipervisor ni contiene un SO invitado. Es una herramienta que permite que los programas escritos en Python se ejecuten en varias CPU.

Al igual que Java, Python traduce sus programas a un formato intermedio que se llama bytecode y los almacena en un archivo listo para su ejecución. Cuando se ejecuta el programa, la VM Python traduce el bytecode en código de máquina para una ejecución rápida.



## Máquinas virtuales Linux

Linux es un SO invitado típico que se usa en muchas VM. También es un SO de host típico que se emplea para ejecutar VM e incluso tiene su propio hipervisor, la máquina virtual basada en kernel (KVM). Aunque es un proyecto de código abierto, Red Hat es propietaria de KVM.

## Máquinas virtuales Ubuntu

Ubuntu es una distribución de Linux producida por Canonical. Está disponible en versiones de escritorio y servidor, que puede instalar como una VM. Los usuarios pueden desplegar Ubuntu como SO invitado en Microsoft Hyper-V. Proporciona una versión optimizada de Ubuntu Desktop que funciona bien en el modo de sesión mejorado de Hyper-V, lo que proporciona una estrecha integración entre el host de Windows y la VM Ubuntu.

Incluye soporte para la integración del portapapeles, el cambio dinámico del tamaño del escritorio, las carpetas compartidas y el movimiento del mouse entre los escritorios host e invitado.

## ¿Qué es un hipervisor o Virtual Machine Monitor (VMM)?

Es un software que agrupa recursos informáticos, como el procesamiento, la memoria y el almacenamiento, y los asigna entre las máquinas virtuales. Esta tecnología posibilita la virtualización, lo que significa que puedes crear y ejecutar varias máquinas virtuales desde una única máquina física.

Los hipervisores también se conocen como supervisores de máquinas virtuales (VMM). Imagínatelos como los supervisores a cargo de repartir los elementos que componen las máquinas virtuales. Toman los recursos de los sistemas de hardware físicos y los incorporan a varias máquinas de una vez, lo que permite crear otras nuevas y gestionar las actuales. Cuando el sistema de hardware físico se usa como hipervisor, se denomina "host", y las múltiples máquinas virtuales que utilizan sus recursos se denominan "guests".

## Hipervisores más utilizados

Los hipervisores más utilizados son los tipo 1 y 2 (Ya vistos en un tema anterior). Aquí uno de los ejemplos de este tipo de visores:

- **VMware vSphere (ESXi):** Un estándar en la industria para centros de datos y virtualización empresarial.
- **Microsoft Hyper-V:** Es la solución de virtualización de Microsoft, muy utilizada en entornos empresariales.
- **KVM (Kernel-based Virtual Machine):** Es un hipervisor de código abierto integrado en el kernel de Linux, muy popular en nubes y entornos de código abierto.

- **Proxmox VE:** Una plataforma de virtualización de código abierto que combina KVM y contenedores, ofreciendo una solución completa para servidores.
- **Xen:** Otro hipervisor de código abierto potente y flexible, utilizado en grandes servicios de la nube.
- **Oracle VM VirtualBox:** Un hipervisor de código abierto gratuito y muy popular para ejecutar máquinas virtuales en Windows, macOS y Linux.
- **Parallels Desktop:** Un hipervisor comercial muy utilizado para ejecutar sistemas operativos Windows y Linux en equipos Mac.
- **VMware Workstation Player:** Similar a VirtualBox, es una solución comercial para usuarios de Windows y Linux que necesiten ejecutar múltiples sistemas operativos en sus equipos.

## Instalación de Hipervisor

Para esta práctica voy a instalar el hipervisor llamado Oracle VirtualBox. Oracle VirtualBox, es el hipervisor 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.

La razón por la que he elegido este hipervisor es porque es el más sencillo de instalar. No te pide una licencia en caso de que se use para estudios o uso personal. El único problema que veo es que suele tener a veces problemas de incompatibilidad.

Para descargarlo, basta con ir a la página de Oracle VirtualBox e ir a la sección de descargas, ahí seleccionar la versión del sistema operativo que quieres.

Una vez instalado, entonces se procede a buscar un ISO de un sistema operativo para su prueba. En este caso serán 3 sistemas operativos los que se utilizarán.

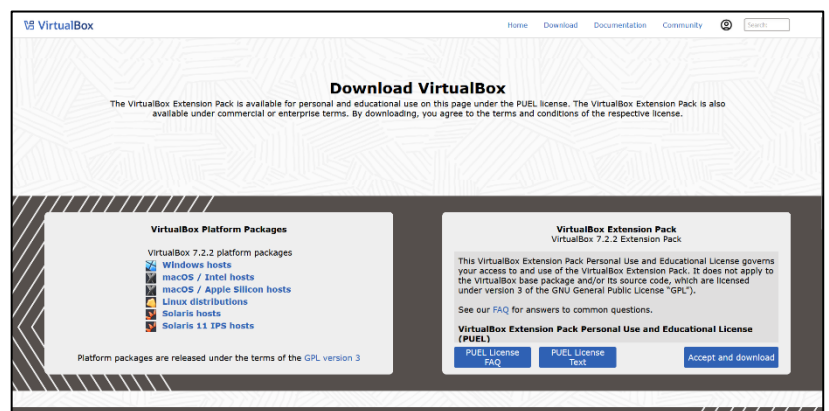


Imagen 1. Descarga de VirtualBox

## Instalación de sistemas operativos

Ahora, para esta actividad he decidido utilizar 3 sistemas operativos con diferencias en su instalación entre ellos. Tratare de mostrar un sistema operativo de base, como lo seria Windows 11 en mi caso, un sistema operativo en dual boot, en este caso Linux con la distribución Pop!\_OS y un sistema operativo con una maquina virtual, en este caso Debian.

### Sistema operativo de Fabrica

Como dije anteriormente, Windows 11 es mi sistema operativo de fábrica. Por eso es que no lo voy a instalar, pero si voy a enseñar como hacerlo.

#### Instalación de Windows 11

En caso de que se quiera descargar este sistema operativo, es necesario ir a la página de Windows y buscar el creador de medios físicos de Windows. Este instalador te permitirá instalar Windows en una memoria USB para usarlo en otra computadora. En caso de que se instale, cabe recalcar que se seguirá necesitando ingresar una licencia original de Windows 11.



Imagen 2. Página de descarga de Windows 11

### Sistema operativo en Dual Boot

El dual boot permite a los usuarios tener dos sistemas operativos en una misma máquina y un mismo dispositivo de almacenamiento. Para la selección de arranque de sistemas operativos existen varias herramientas que se pueden instalar desde Linux, como **GRUB** o **Systemd-boot** (En mi caso). Gracias a esto he podido instalar Pop!\_OS en un disco a parte en mí misma computadora.

Para instalar Pop!\_OS, es necesario ir a su pagina web para descargar el ISO del S.O. Existen varias versiones para descargar, la primera es una versión base sin ninguna característica añadida. La segunda es una versión que viene compilada con los drivers de NVIDIA, lo cual esta bien para usar en computadoras con tarjetas gráficas.

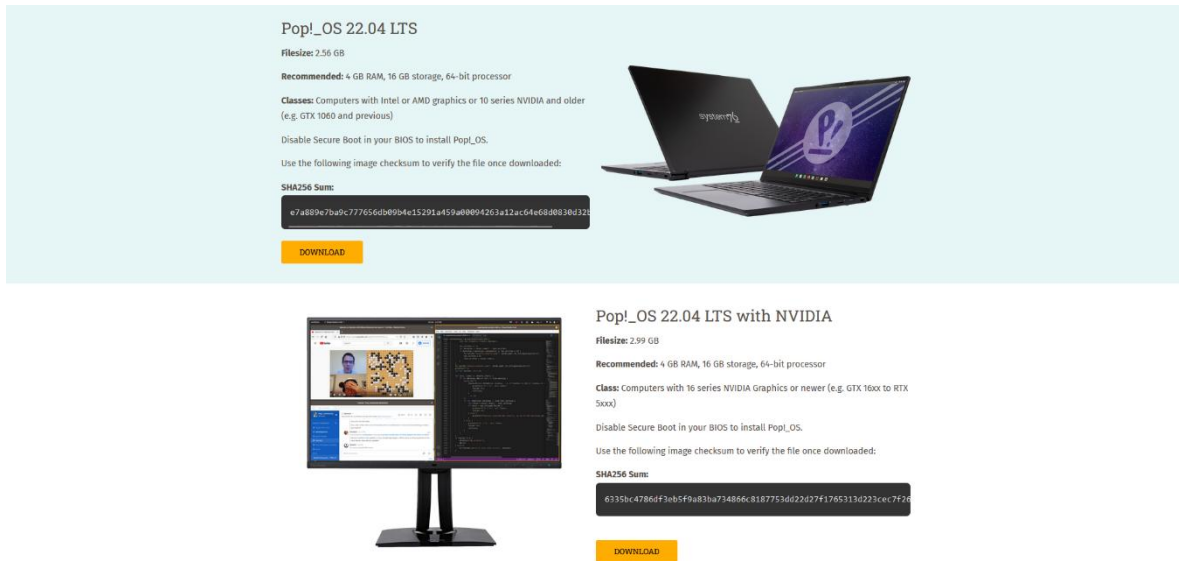


Imagen 3. Versiones de Pop!\_OS

Una vez descargado, es necesario instalarlo en una memoria a través de Rufus. Es necesario también que hagas particiones en el disco duro de tu máquina. En mi caso le di mas de 200GB de espacio a Linux, esto para que tuviera el espacio suficiente para actualizaciones y programas.

Disco 0 Básico 894.24 GB En línea	SSD 1TB (D:) 644.24 GB NTFS Correcto (Partición de datos)	1.00 GB Correcto (Partición de datos)	4.00 GB No asignado	8.00 GB Correcto (Partición de datos)	100.00 GB Correcto (Partición primaria)	137.00 GB Correcto (Partición primaria)
	300 MB Correcto (Partición de sistema)	Windows (C:) 237.08 GB NTFS Correcto (Arranque, Archivo de paginación, Volcado, Partición de recuperación)	996 MB Correcto (Partición de recuperación)			
Disco 1 Básico 238.35 GB En línea						

■ No asignado ■ Partición primaria

Imagen 4. Particiones de mi disco duro

También es visible que tengo los dos sistemas operativos en diferentes discos.

## Sistema operativo en máquina virtual

Para terminar el sistema operativo en maquina virtual, para este ejemplo utilizaremos Debian. Debian es el sistema operativo que muchos expertos en redes utilizan para montar sistemas operativos. Para su descarga es necesario ir a su página web.



Imagen 5. Página web de Debian

## Instalación de Debian en Máquina Virtual

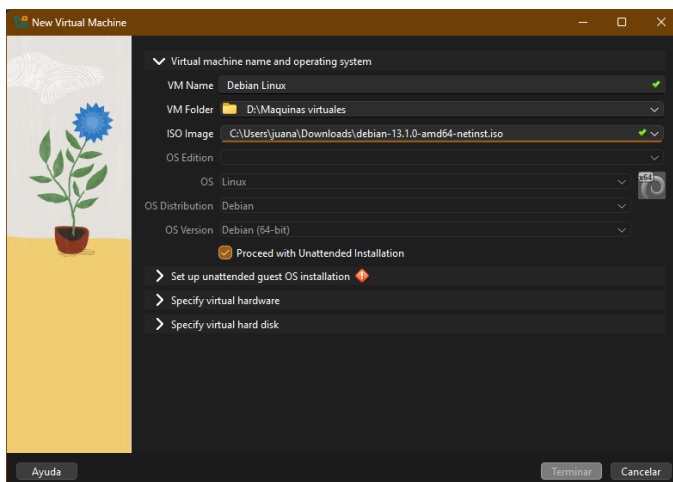


Imagen 6. Configuración de Debian

A diferencia de una instalación en máquina, una máquina virtual necesita el ISO para instalarse desde el mismo programa. Con el ISO que descargamos, instalaremos en VirtualBox. Crearemos una nueva máquina virtual, dándole la ruta donde se instalara y el ISO seleccionado, en este caso, Debian.

Una vez creada, debería verse así:

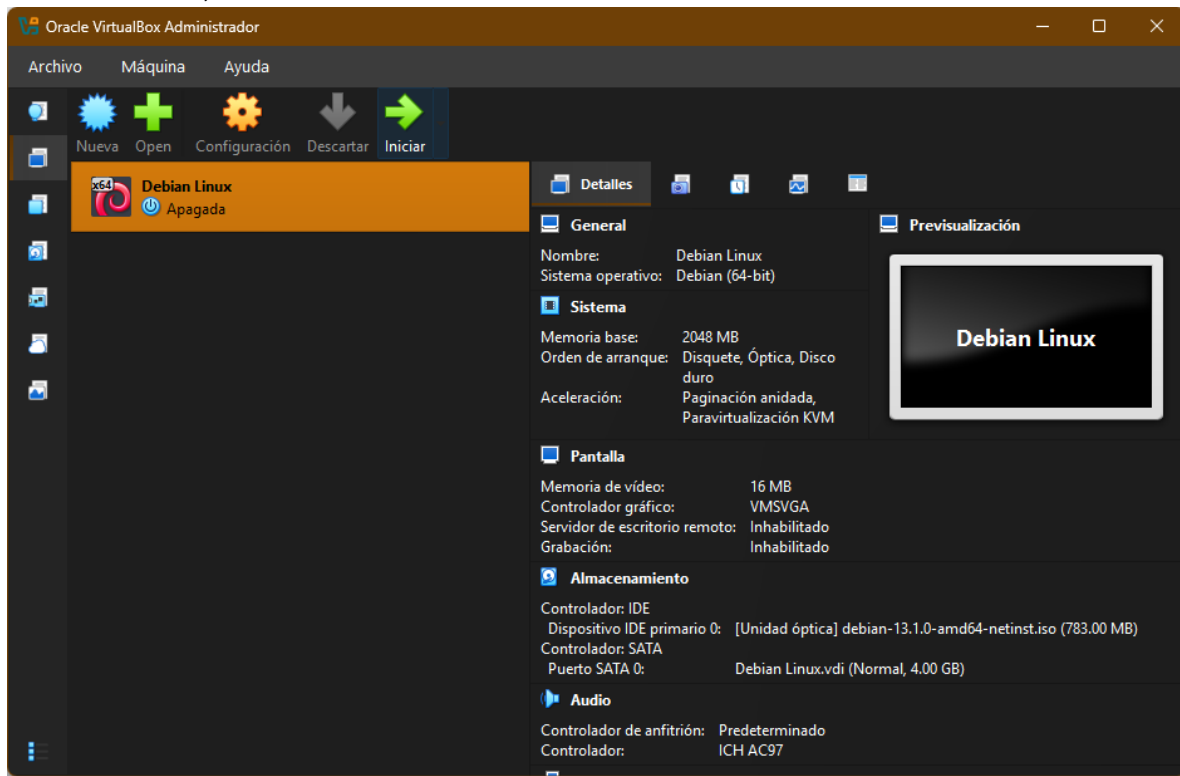


Imagen 7. Máquina virtual creada

Procederemos a prenderla para iniciar el proceso de instalación.

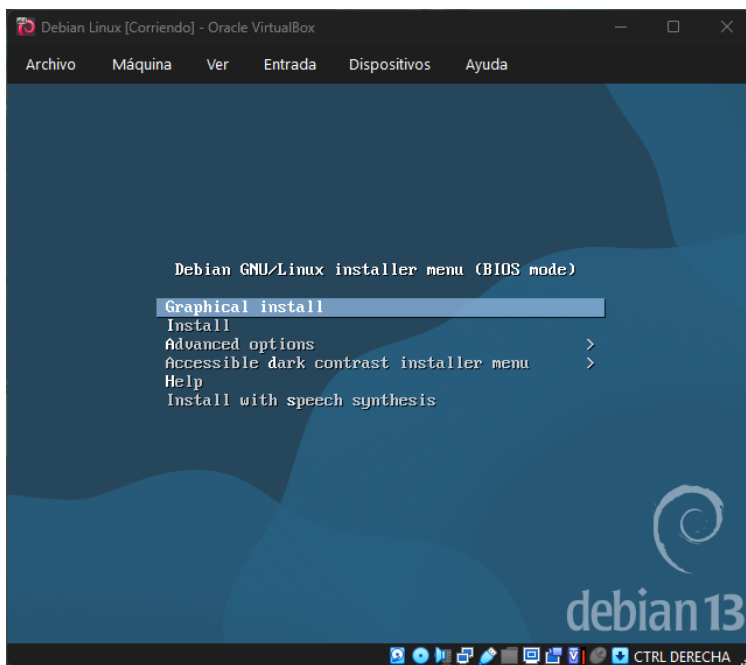
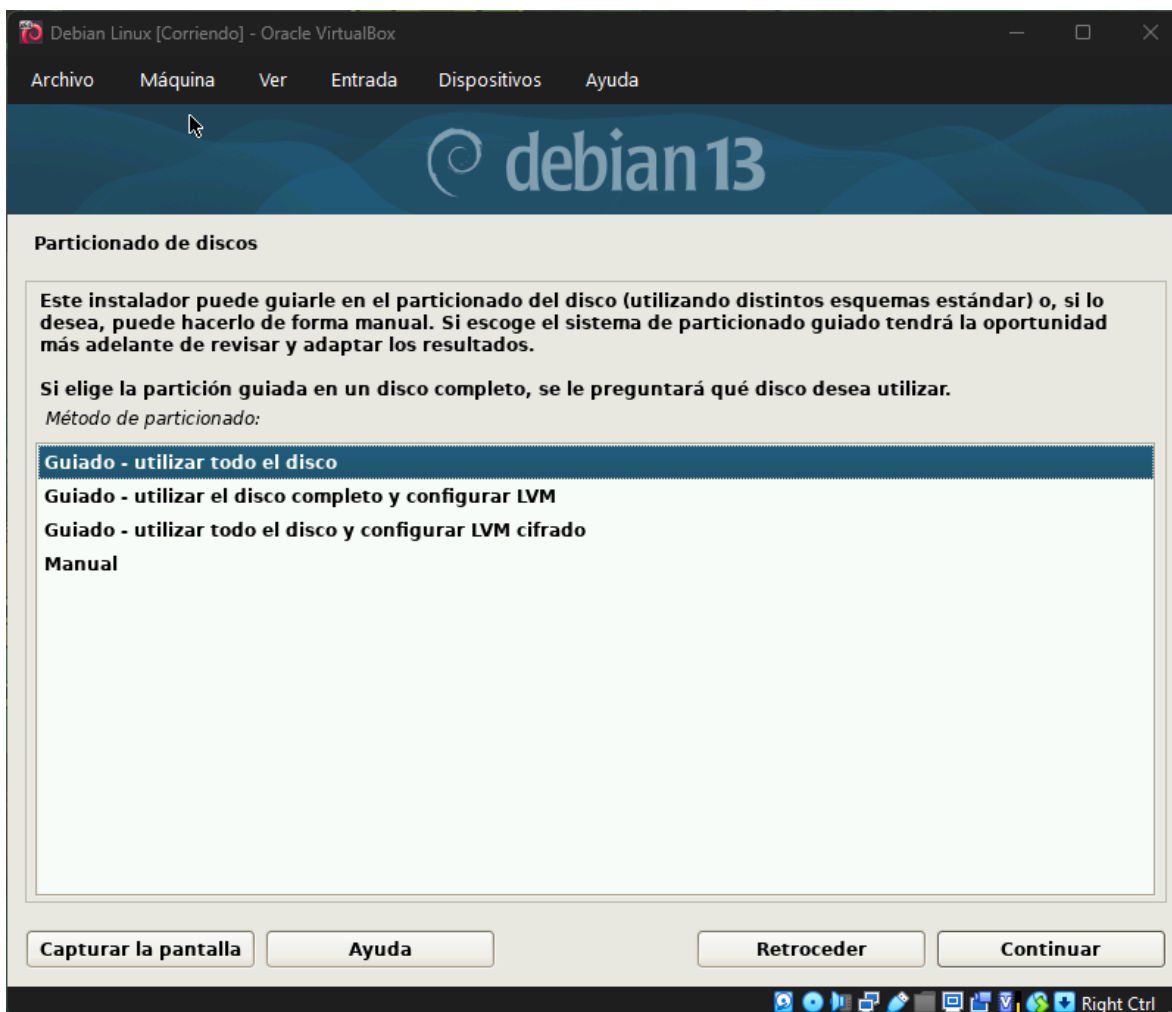


Imagen 8. Configuración de Instalación Debian

En opciones elegiremos nuestro idioma y región, esto establecerá el idioma del sistema operativo y la distribución del teclado. Luego nos pedirá como utilizar el disco de Debian, le daremos a “Guiado – utilizar todo el disco”. En el siguiente paso pondremos todo en una sola partición.



*Imagen 9. Uso del disco en Debian*

Seleccionaremos luego los paquetes que son de Debian. Esto nos permite acceder al repositorio y gestor de paquetes de Debian. Cuando terminemos la instalación terminara de verse así el arranque:

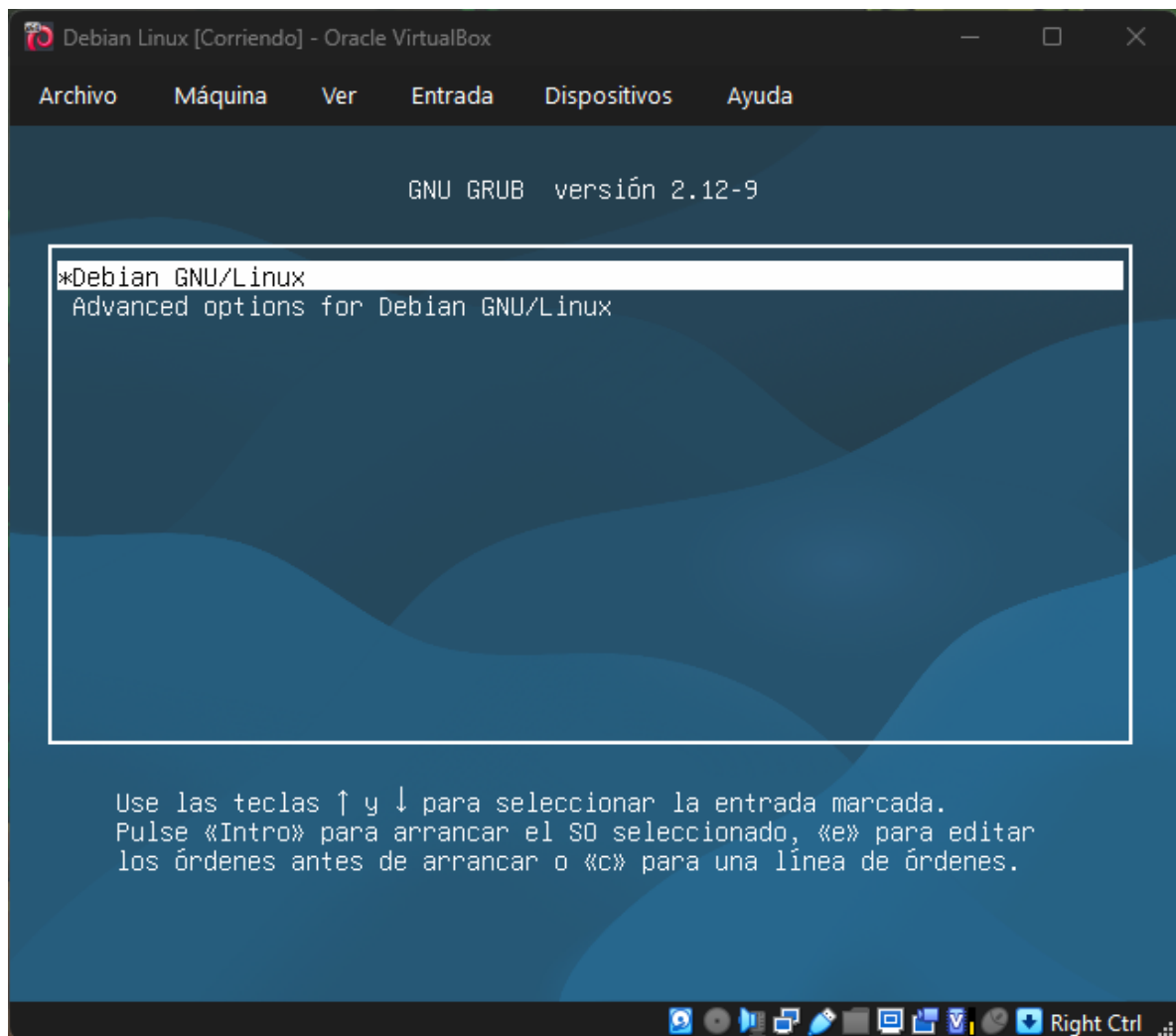


Imagen 10. Arranque de Debian

Le damos a Debian GNU/Linux y iniciamos Debian. Al iniciar se debería verse así.

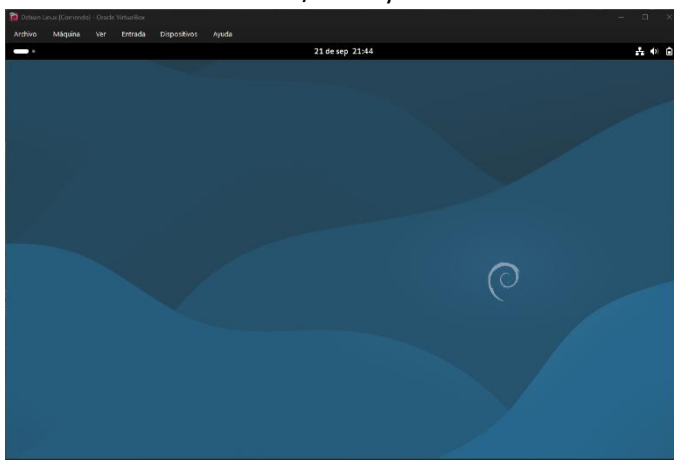


Imagen 11. Sistema operativo Debian

De esta manera tenemos 3 sistemas operativos en una misma computadora.



## Video de demostración

En este video se demuestra el correcto funcionamiento de los 3 sistemas operativos en la computadora:

[https://drive.google.com/file/d/1yAxE-PAUc3sRXXpBUiGY2azsPW\\_1eMmM/view?usp=sharing](https://drive.google.com/file/d/1yAxE-PAUc3sRXXpBUiGY2azsPW_1eMmM/view?usp=sharing)

## Conclusiones

Gracias a esta actividad, comprendí mejor la diferencia entre un hipervisor y una máquina virtual. Me pareció fascinante cómo un sistema operativo completo puede operar de forma encapsulada y abstraída dentro de un programa. Esta tecnología es ideal para probar un nuevo sistema operativo, ya que permite experimentar sin el riesgo de alterar las configuraciones de nuestra computadora principal.

Por otro lado, también aprendí que la virtualización puede ser un proceso delicado. Mi propia experiencia fue un claro ejemplo de esto, ya que tuve problemas para iniciar VirtualBox debido a un programa que modificaba la apariencia de Windows. Esto me demostró cuántos factores externos pueden interferir con el correcto funcionamiento de una máquina virtual.

## Referencias

Susnjara, S., & Smalley, I. (2025, 7 mayo). Máquinas virtuales. *IBM*. Recuperado 21 de septiembre de 2025, de <https://www.ibm.com/mx-es/think/topics/virtual-machines>

Oracle VirtualBox. (s. f.). <https://www.oracle.com/mx/virtualization/virtualbox/>

Hipervisor: ¿qué es, cómo funciona y qué tipos hay? (s. f.). <https://www.redhat.com/es/topics/virtualization/what-is-a-hypervisor>