

ACTIVIDAD #1 EXPLORANDO DISTRIBUCIONES LINUX

SISTEMAS OPERATIVOS

ALUMNO: CRISTIAN MORALES BERRIO

DOCENTE: ING. JAIDER REYES HERAZO

Ingeniería de Sistemas

Antonio Jose De Sucre

2025

ACTIVIDAD #1 - DISTRIBUCIONES LINUX - Explorando openSUSE MicroOS

1. Ficha Técnica

A continuación, presento un resumen con las especificaciones técnicas de la distribución analizada:

Campo	Detalle
Nombre	openSUSE MicroOS
Año de lanzamiento	2018 (es un derivado de openSUSE Tumbleweed)
Última versión estable	Es de tipo <i>Rolling Release</i> continuo, por lo que no tiene un número de versión fijo.
Tipo de soporte	Rolling Release / Inmutable
Entorno de escritorio	Viene sin escritorio por defecto (<i>headless</i>), aunque permite usar GNOME opcionalmente a través de MicroOS Desktop.
Requerimientos mínimos	CPU: x86-64/ARM; RAM: 1 GB mínimo (se recomiendan 2 GB); Almacenamiento: 20 GB.
Enfoque principal	Diseñado para sistemas de contenedores, servidores, <i>edge computing</i> e infraestructura inmutable.

¿Qué hace especial a openSUSE MicroOS?

Lo que más me llamó la atención en el laboratorio es que esta distribución rompe con el esquema tradicional: su sistema de archivos raíz es de solo lectura mientras está en ejecución. Esto implica que no se puede modificar el sistema operativo mientras corre, dándole una estabilidad enorme frente a errores humanos o fallos de actualización.

Las actualizaciones se manejan de manera transaccional con la herramienta transactional-update, apoyándose en Snapper y el sistema de archivos Btrfs. En la práctica, cada vez que se actualiza algo, el sistema toma una "foto" o *snapshot*. Si la actualización daña el sistema, basta con reiniciar y cargar el *snapshot* anterior. Es básicamente un modelo de "actualiza sin miedo, que si falla, se revierte solo".

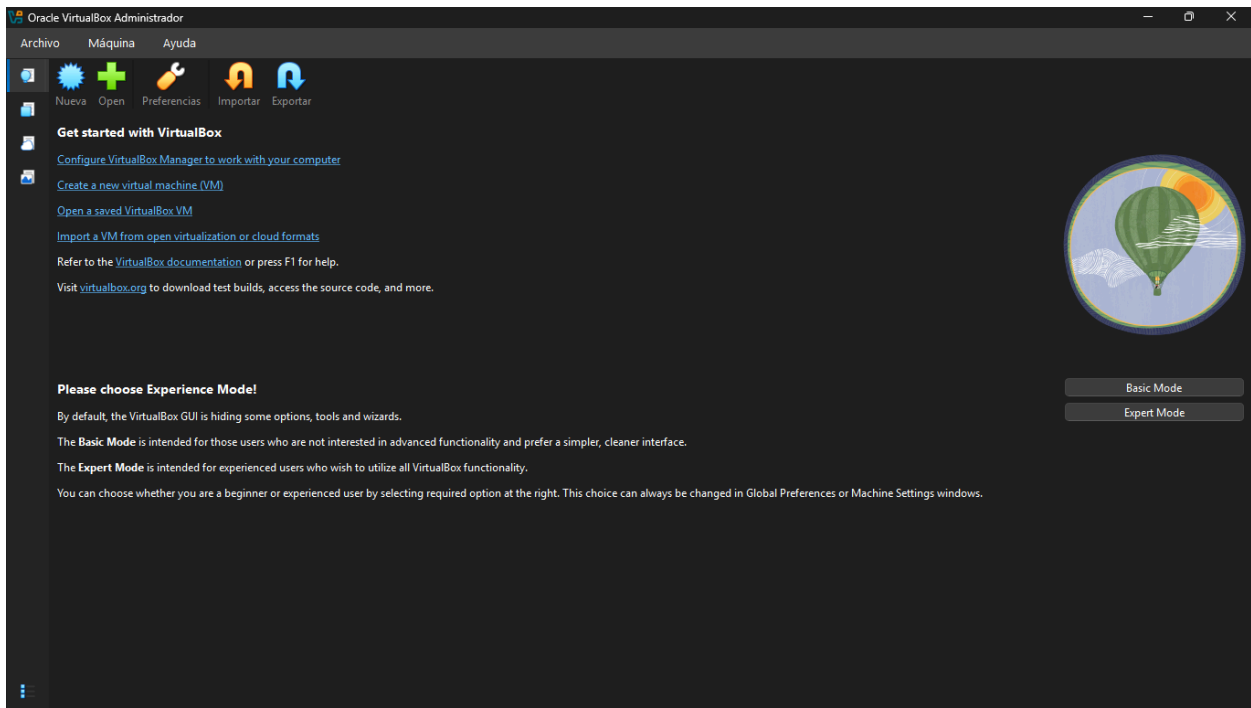
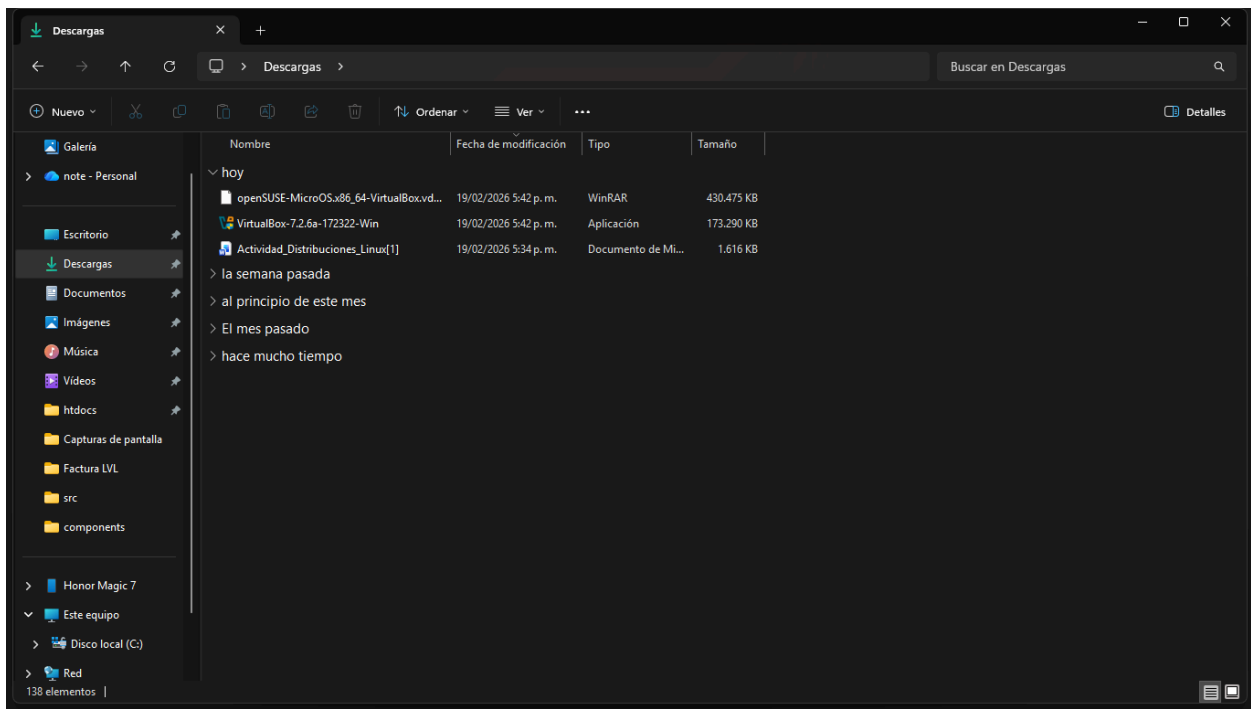
2. Instalación en Máquina Virtual

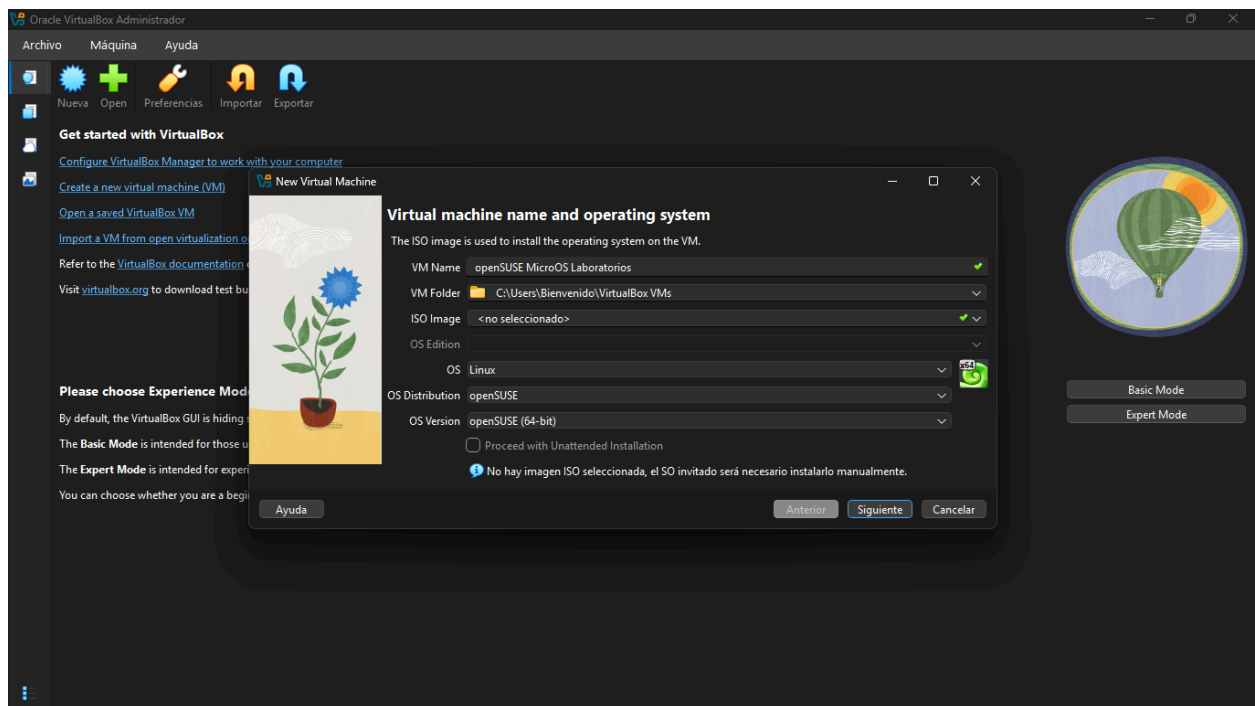
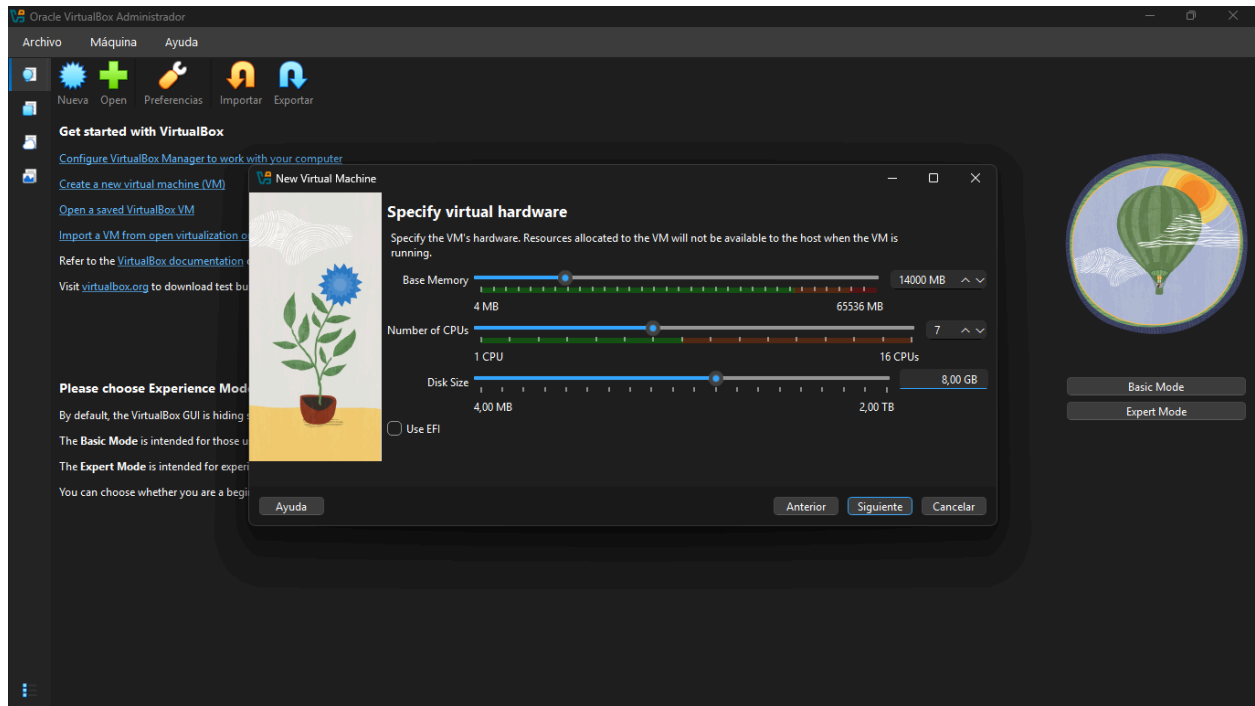
Para llevar a cabo la práctica, configuré un entorno virtualizado con los siguientes parámetros:

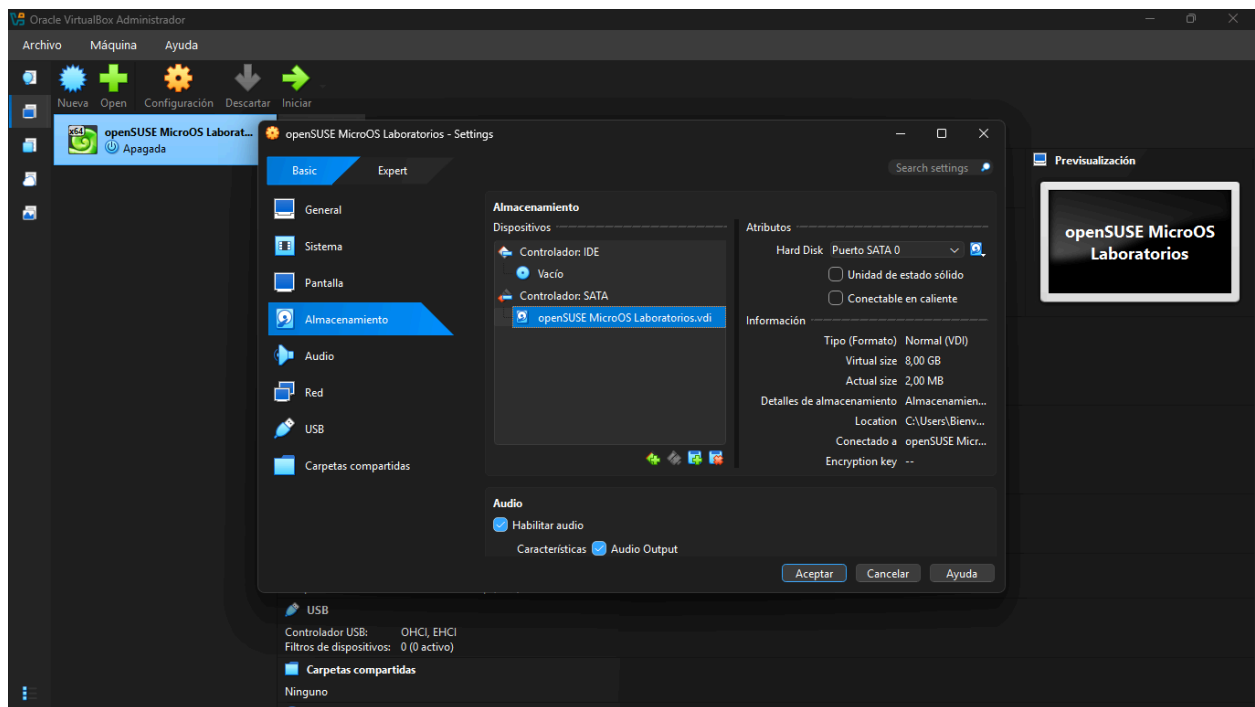
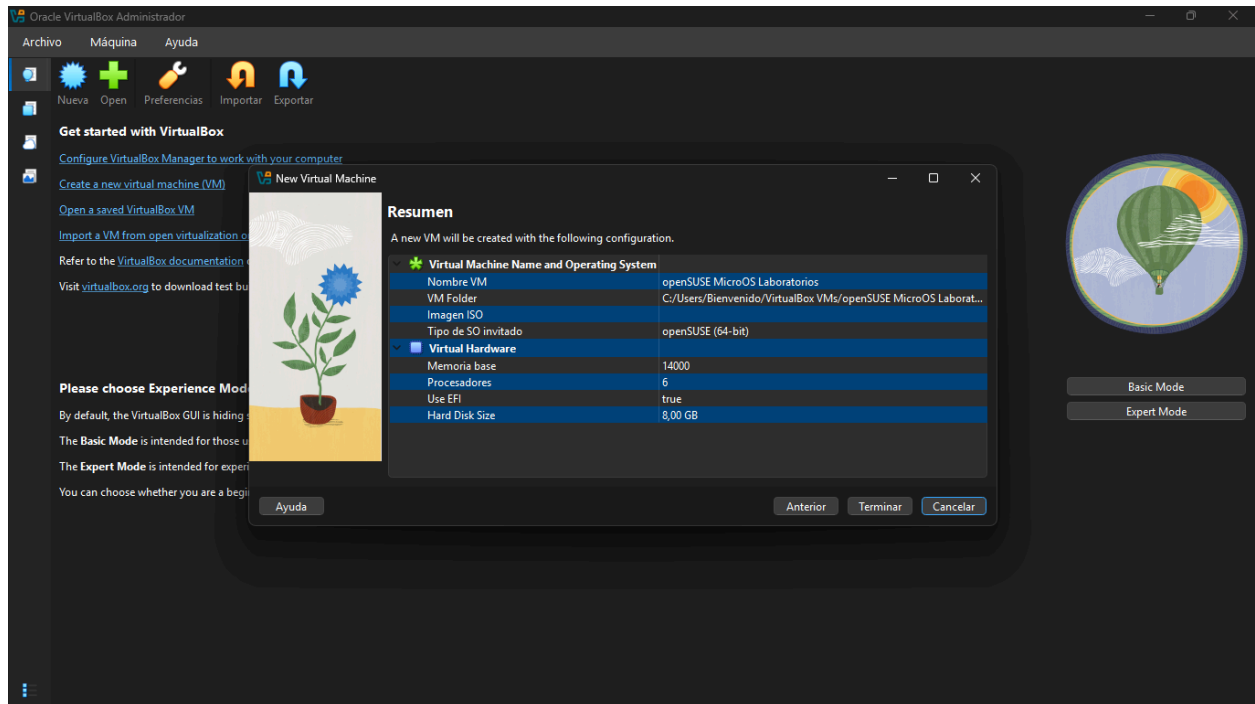
- **Herramienta utilizada:** VMware Workstation / VirtualBox (*Nota: borra el que no hayas usado*)
- **ISO descargada desde:** <https://microos.opensuse.org/>
- **RAM:** 2 GB
- **Procesadores:** 2 núcleos
- **Disco:** 20 GB (asignación dinámica)
- **Tipo de sistema:** Linux, openSUSE (64-bit)

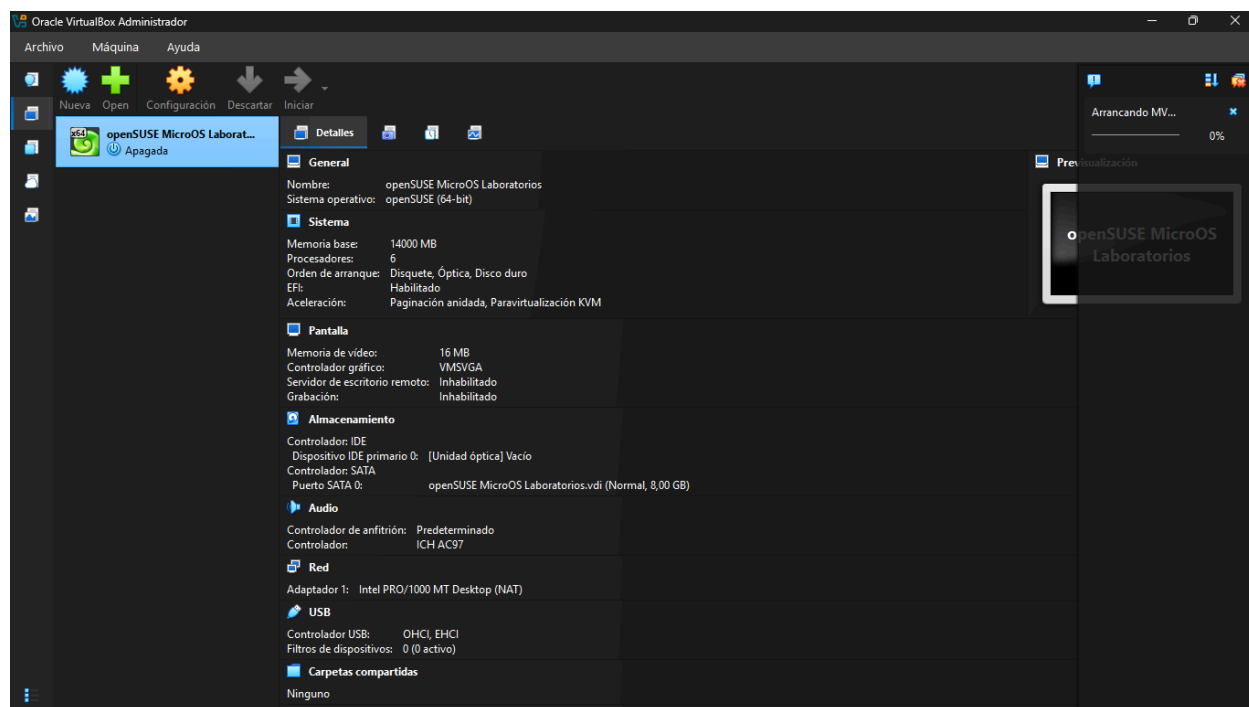
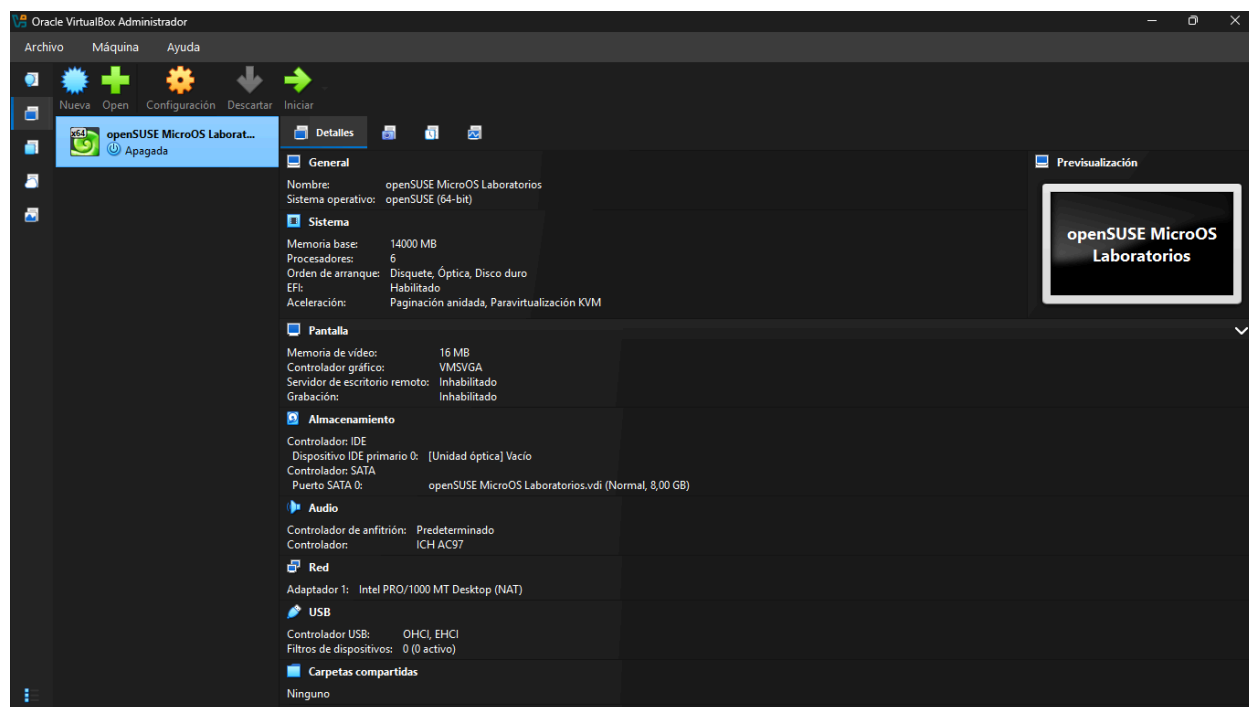
Durante la instalación noté que openSUSE MicroOS utiliza YaST, el conocido instalador gráfico de la familia openSUSE. Seleccioné el perfil "Container Host", ya que es el propósito principal para el que fue diseñado este SO, dejando de lado la opción de escritorio con GNOME. Un punto clave de la instalación es que YaST se encarga de configurar automáticamente todo el esquema de particiones Btrfs con subvolumenes y la herramienta Snapper, preparando el terreno para las actualizaciones transaccionales.

Evidencias del proceso de instalación:









3. Prueba Práctica en Terminal

Una vez instalado, procedí a ejecutar comandos básicos para familiarizarme con el entorno y comprobar el funcionamiento del sistema.

- **Listar directorios (ls):** Muestra los archivos y carpetas en la ubicación actual. Es fundamental para movernos por el sistema.
- **Ver directorio actual (pwd):** *Print Working Directory* me sirvió para confirmar la ruta absoluta en la que estaba trabajando dentro del árbol de directorios.
- **Cambiar de directorio (cd):** Lo utilicé para navegar a otras rutas, por ejemplo, entrando a `/etc` para buscar archivos de configuración.
- **Leer archivos (cat):** Me permitió visualizar el contenido de archivos de texto directamente en consola. Lo probé leyendo `/etc/os-release` para ver los datos internos de la distribución.

Instalación de paquetes (El paradigma inmutable)

Esta fue la parte más diferente de la práctica. En sistemas como Ubuntu usaríamos `apt install`, pero como la raíz de MicroOS es de solo lectura, eso da error. Se debe usar `transactional-update pkg install [paquete]`. Esto crea un *snapshot*, instala el paquete por debajo usando zypper (el gestor nativo de openSUSE) y requiere un reinicio para que los cambios apliquen.

4. Estructuras, Elementos y Características del Sistema Operativo

Para comprender cómo funciona este SO, es útil visualizar la arquitectura de Linux en capas (desde el hardware físico hasta las aplicaciones del usuario). La principal diferencia de MicroOS frente a sistemas como Debian es la capa adicional del sistema de archivos transaccional.

- **Kernel:** Es el núcleo que habla con el hardware físico (CPU, RAM, discos). Como esta distro se basa en Tumbleweed, trae un kernel muy reciente, asegurando buena compatibilidad. Verifiqué mi versión ejecutando `uname -r`.
- **Shell:** Es el intérprete de los comandos que escribo en la terminal. Confirmé que MicroOS usa Bash por defecto mediante el comando `echo $SHELL`.
- **Funciones del Sistema Operativo en la práctica:**
 - **Procesos:** El sistema usa `systemd` como proceso padre (PID 1). Pude ver la lista de procesos activos con `ps aux`.
 - **Memoria:** El kernel asigna la RAM dinámicamente. Verifiqué el consumo de la máquina virtual con `free -h`.
 - **Almacenamiento:** Controlado por Btrfs y Snapper. Usé `df -h` para ver el espacio y `sudo snapper list` para ver los *snapshots* generados por el sistema.
 - **Dispositivos (E/S):** El kernel administra el hardware conectado. Listé mis discos virtuales con `lsblk`.

5. Proceso de Arranque (Bootloader)

- **Bootloader utilizado:** GRUB 2 (aunque soporta `systemd-boot` en hardware UEFI muy moderno).

El gestor de arranque es lo primero que se ejecuta tras pasar el chequeo de la BIOS/UEFI. Sus tareas principales son buscar el kernel en el disco, subirlo a la RAM, pasarle parámetros y finalmente cederle el control del equipo.

En MicroOS, el GRUB tiene una misión extra vital: **su menú muestra el historial de *snapshots***.

Gracias a esto, si rompo el sistema con un mal comando, desde la misma pantalla de arranque puedo seleccionar un estado anterior y encender la máquina como si nada hubiera pasado.

También pude observar parte de su archivo de configuración interno leyendo `/boot/grub2/grub.cfg`.

6. Conclusión

Esta práctica con openSUSE MicroOS me permitió entender un paradigma distinto al de las distribuciones clásicas que solemos usar en clase. Comprobé de primera mano que un sistema de archivos "inmutable" no significa que sea un sistema estancado que no recibe actualizaciones; al contrario, es un sistema que se actualiza de una manera mucho más robusta y segura. La integración nativa de Btrfs y Snapper para revertir cambios defectuosos lo convierte en una solución excelente para servidores críticos o sistemas de contenedores. Definitivamente, es una tendencia (similar a lo que hacen Fedora Silverblue o ChromeOS) que seguiremos viendo crecer en la administración moderna de Sistemas Operativos.