

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

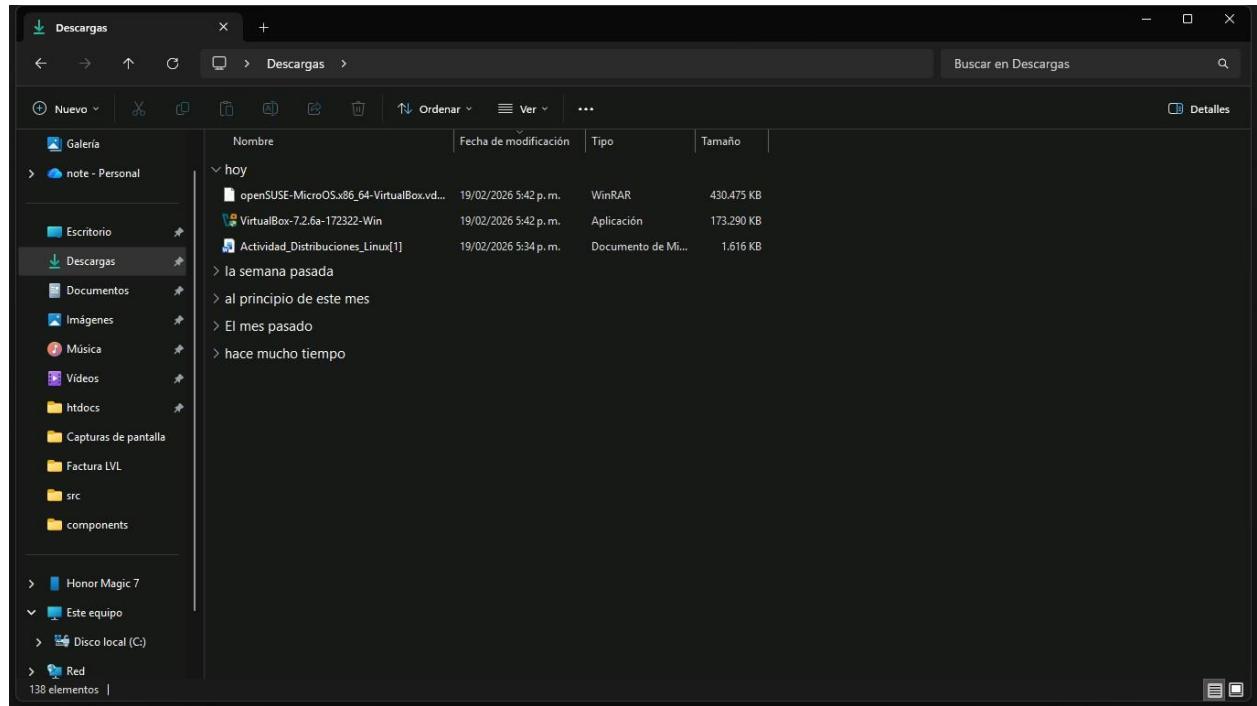
Para llevar a cabo la práctica, configuré un entorno virtualizado con los siguientes parámetros:

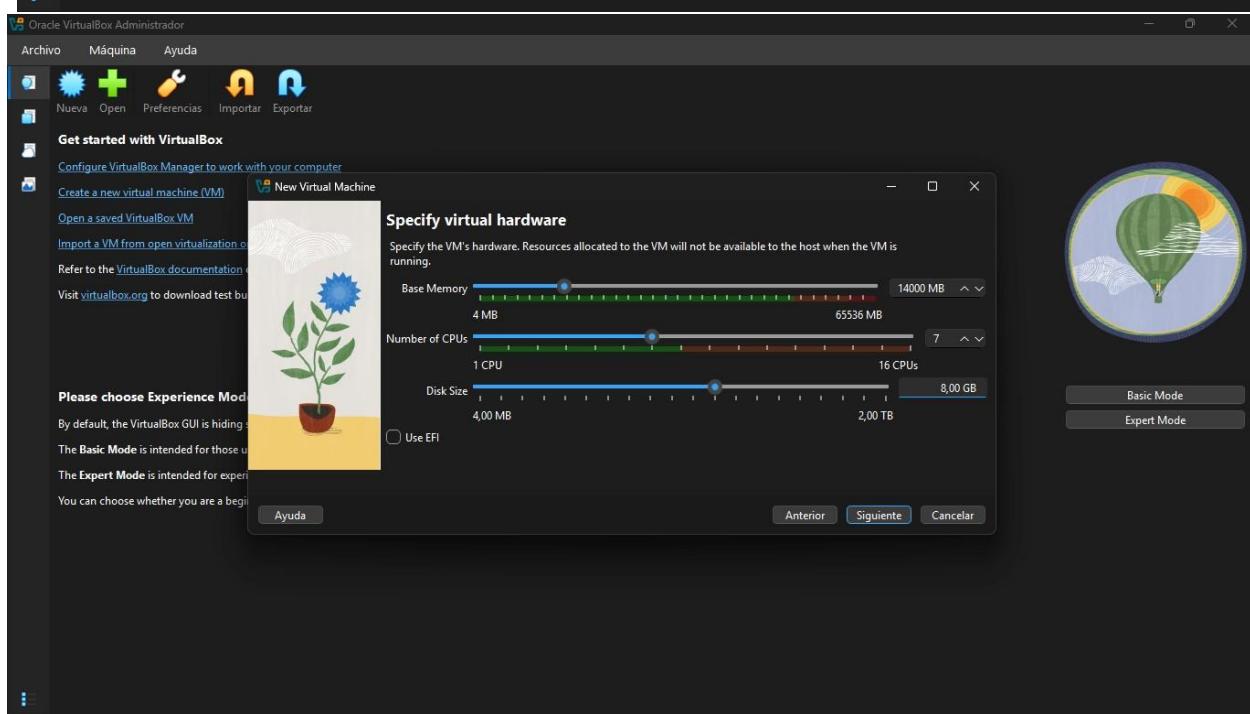
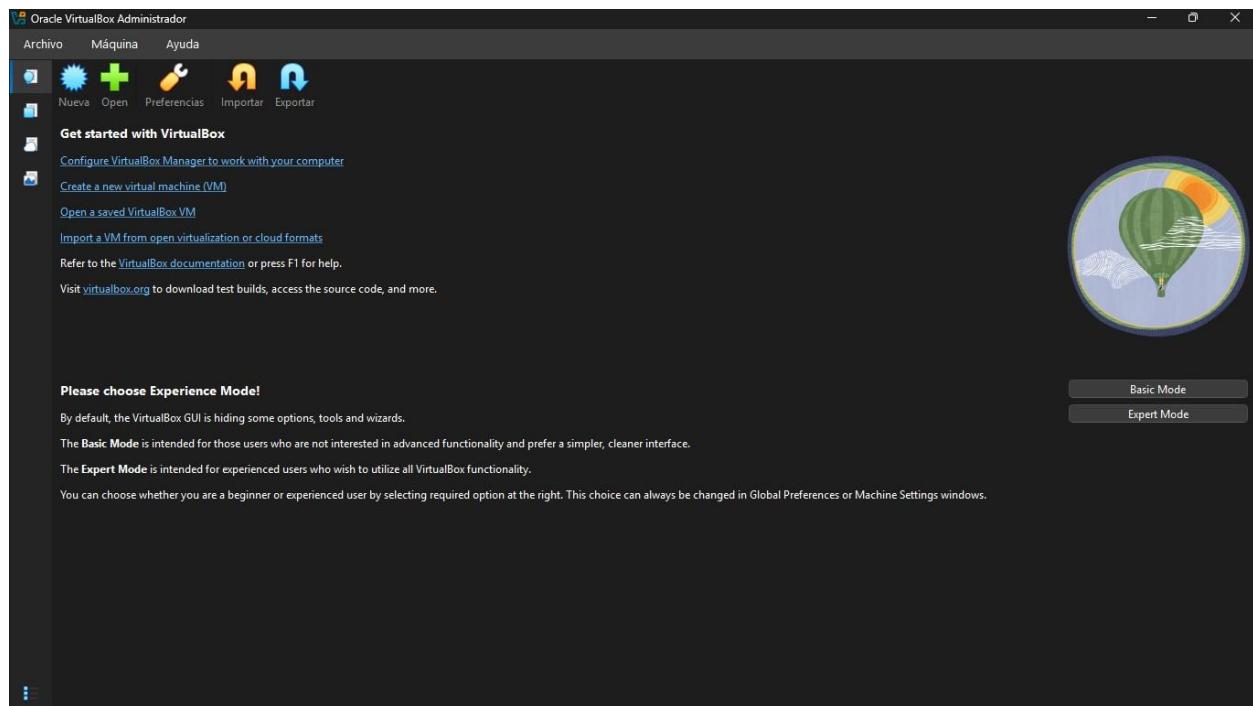
- **Herramienta utilizada:** VMware Workstation / VirtualBox (*Nota: borra el que no hayas usado*)
- **ISO descargada desde:** <https://microos.opensuse.org/>
- **RAM:** 2 GB
- **Procesadores:** 2 núcleos
- **Disco:** 20 GB (asignación dinámica)
- **Tipo de sistema:** Linux, openSUSE (64-bit)

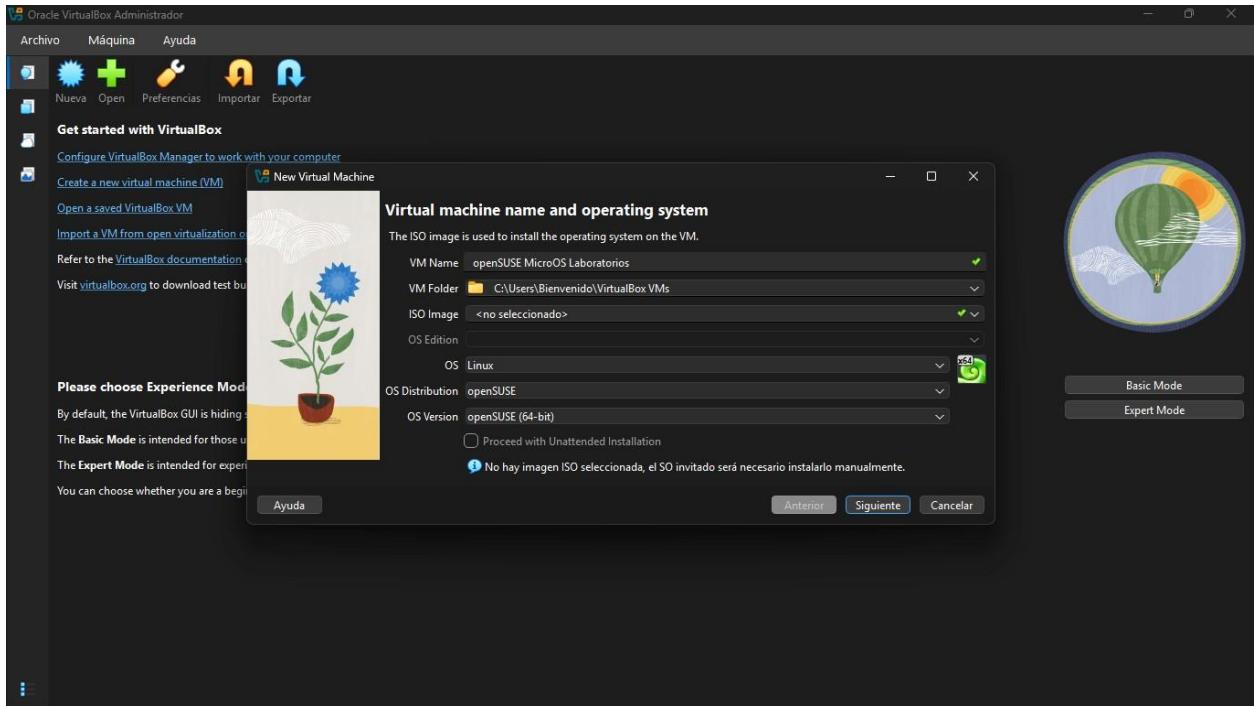
Durante la instalación noté que openSUSE MicroOS utiliza YaST, el conocido instalador gráfico de la familia openSUSE. Seleccioné el perfil "Container Host", ya que es el propósito principal para el que fue diseñado este SO, dejando de lado la opción de escritorio con GNOME. Un punto clave de la instalación

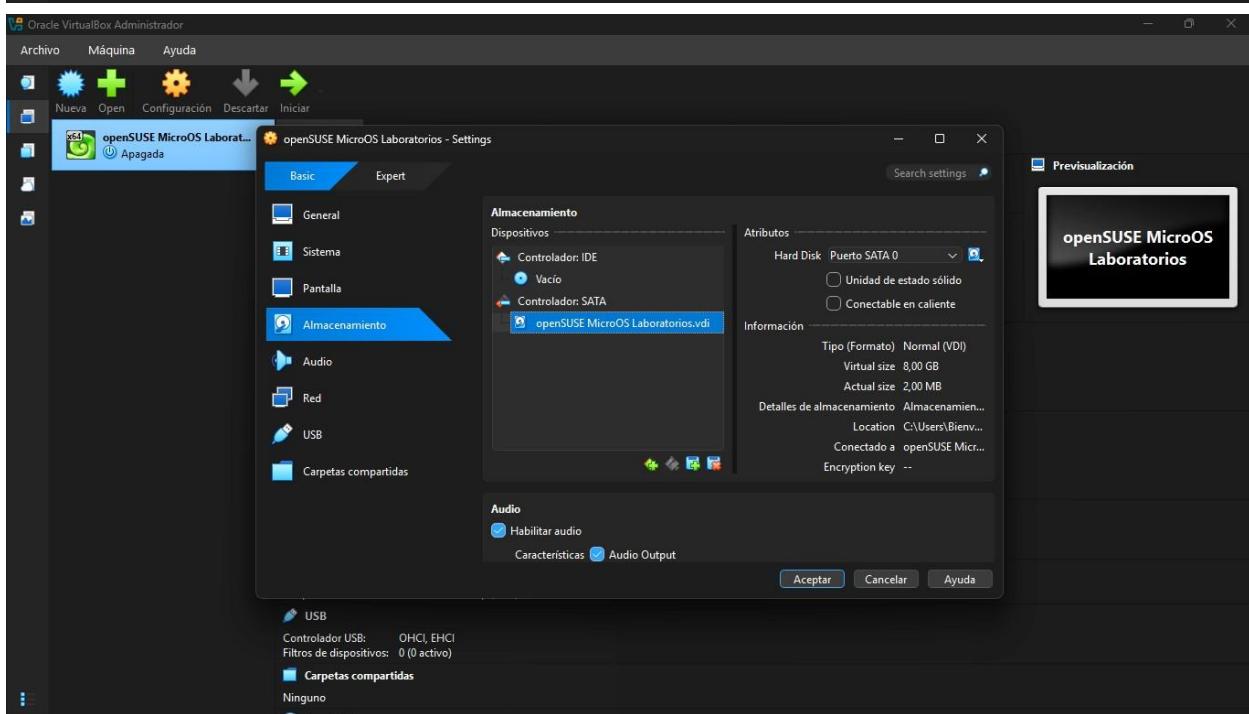
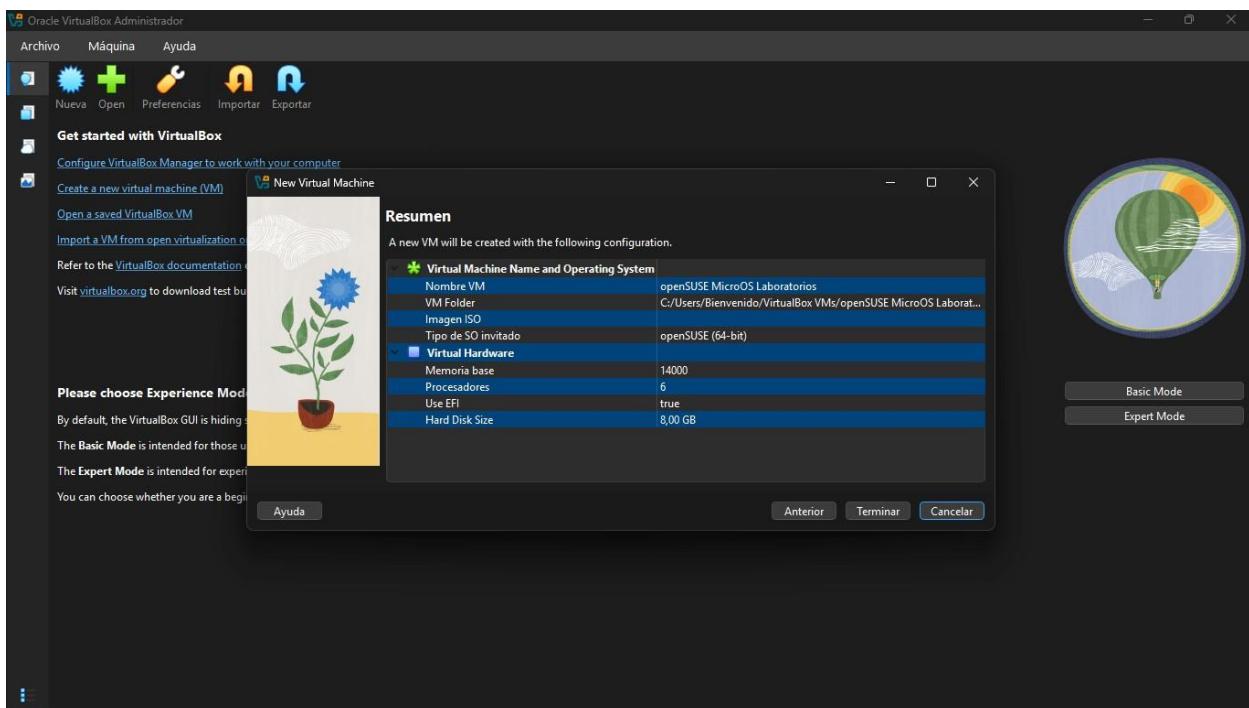
es que YaST se encarga de configurar automáticamente todo el esquema de particiones Btrfs con subvolúmenes y la herramienta Snapper, preparando el terreno para las actualizaciones transaccionales.

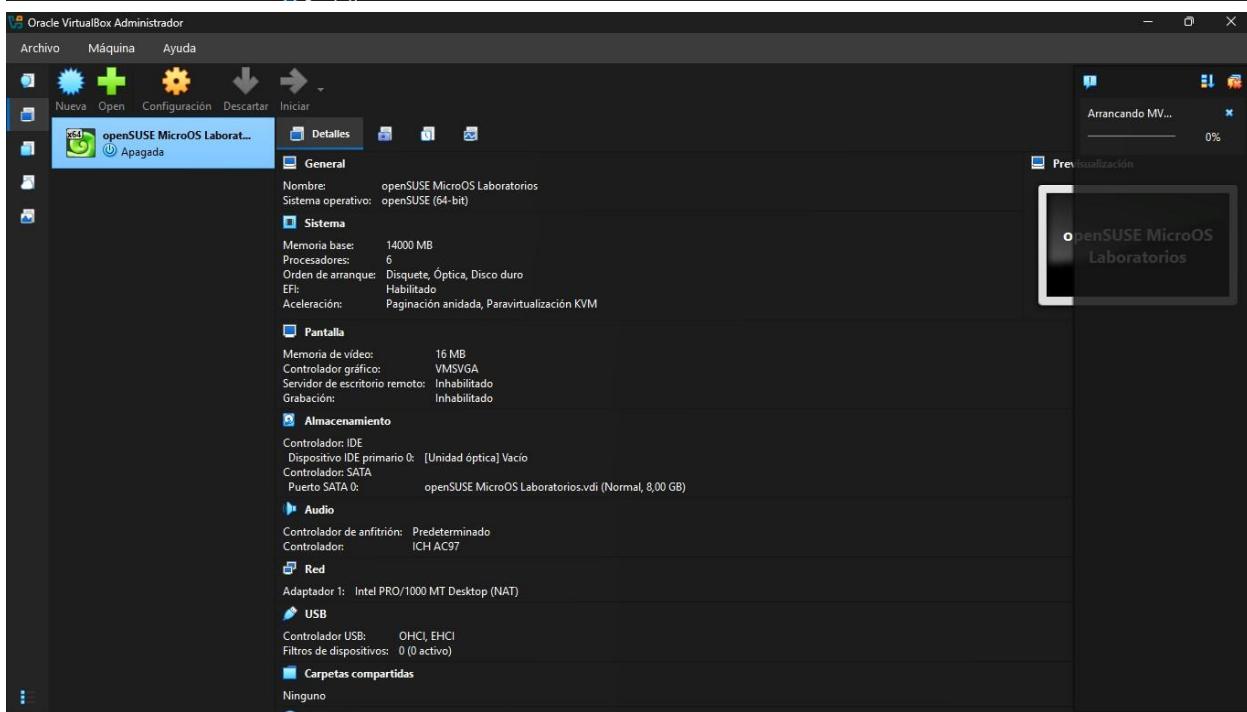
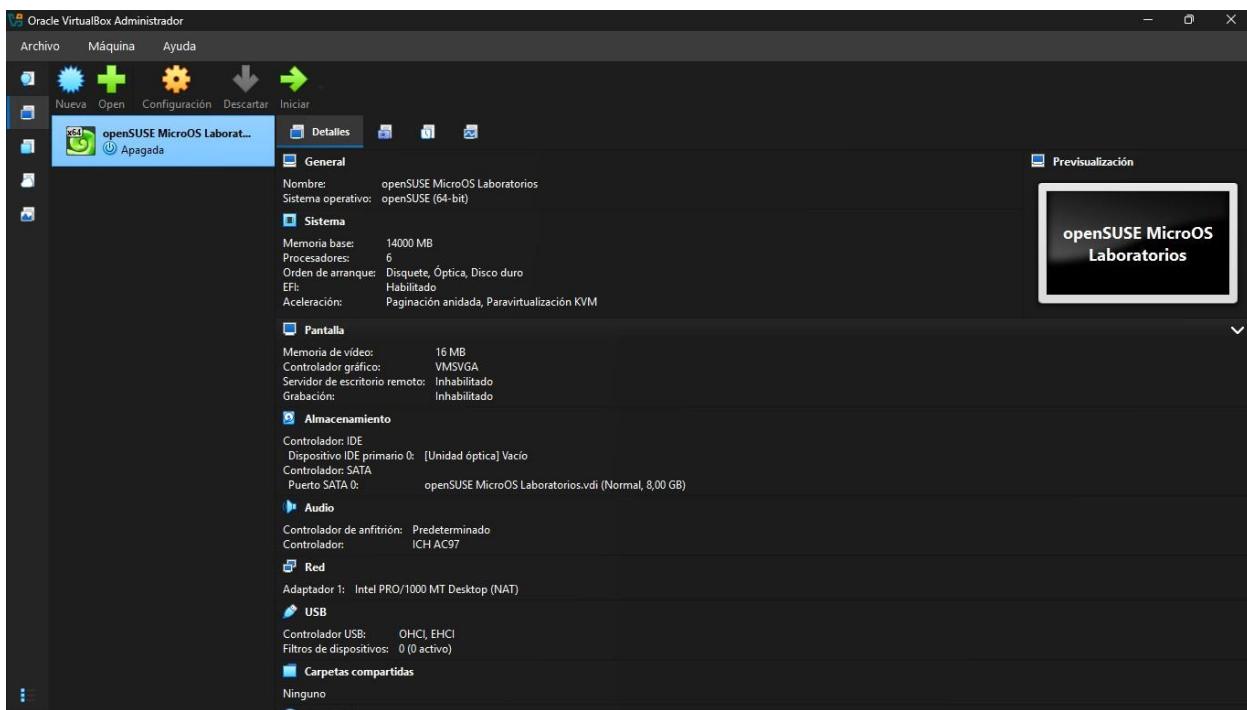
Evidencias del proceso de instalación:

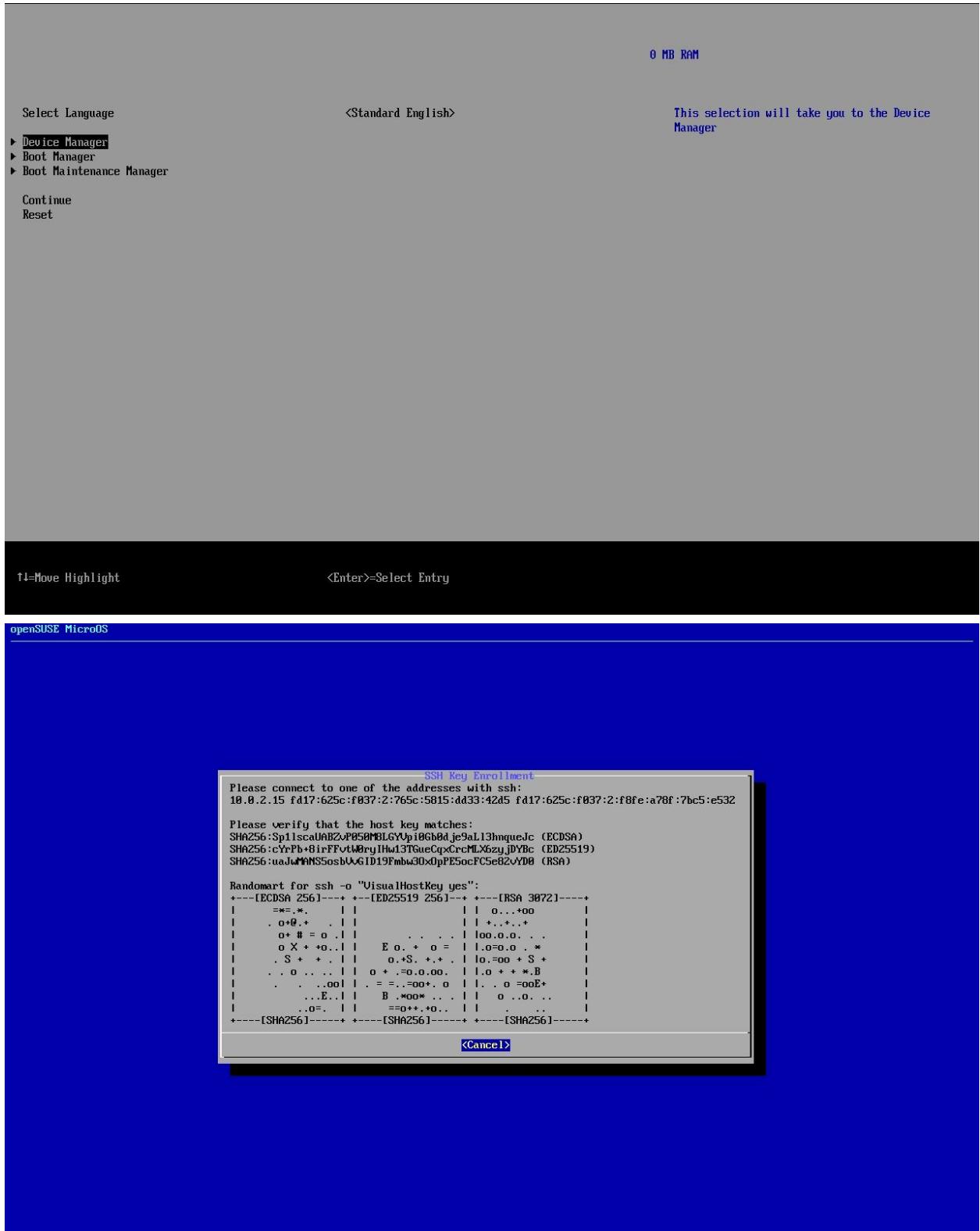












3. Prueba Práctica en Terminal

```
[ OK ] Finished Write boot and shutdown times into wtmpdb.
[ OK ] Started Hostname Service.
[ OK ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
      Starting Network Manager Script Dispatcher Service...
[ OK ] Started Network Manager.
[ OK ] Reached target Network.
      Starting NTP client/server...
      Starting OpenSSH Daemon...
      Starting Permit User Sessions...
[ OK ] Started Network Manager Script Dispatcher Service.
[ OK ] Finished Permit User Sessions.
[ OK ] Started Modem Manager.
[ OK ] Started OpenSSH Daemon.
[ OK ] Started NTP client/server.
[ OK ] Reached target System Time Synchronized.
[ OK ] Started Balance block groups on a btrfs filesystem.
[ OK ] Started Defragment file data and/or directory metadata.
[ OK ] Started Scrub btrfs filesystem, verify block checksums.
[ OK ] Started Discard unused blocks on a mounted filesystem.
[ OK ] Started Discard unused filesystem blocks once a week.
[ OK ] Started Daily rotation of log files.
[ OK ] Started Timeline of Snapper Snapshots.
[ OK ] Started Daily update of the system.
[ OK ] Started Monthly rotation of wtmpdb.
[ OK ] Reached target Timer Units.
[ OK ] Started Getty on tty1.
[ OK ] Started Serial Getty on ttyS0.
[ OK ] Reached target Login Prompts.
[ OK ] Finished MicroOS Health Checker.
[ OK ] Reached target Multi-User System.

Welcome to openSUSE MicroOS (x86_64) - Kernel 6.19.2-1-default (tty1).
enp0s3: 10.0.2.15

localhost login:
Welcome to openSUSE MicroOS (x86_64) - Kernel 6.19.2-1-default (tty1).
enp0s3: 10.0.2.15 fd17:625c:f037:2:ed24:f654:99a:8b82

localhost login: cristian
Password:
Last login: Wed Feb 25 14:04:41 -05 2026 on tty1
cristian@localhost:~>
```

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.

```
cristian@localhost:~> ls  
explicacion_punto1.txt  
cristian@localhost:~> pwd  
/home/cristian  
cristian@localhost:~> cd  
cristian@localhost:~> cat  
uname -r  
uname -r
```

```
clearrr  
clearrr
```

```
clear  
clear
```

```
-
```

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

```
cristian@localhost:~> uname -r  
6.19.2-1-default  
cristian@localhost:~> echo $SHELL  
>/bin/bash  
cristian@localhost:~> echo $XDG_CURRENT_DESKTOP  
cristian@localhost:~> pacman --version  
-bash: pacman: command not found  
cristian@localhost:~> pacman --version  
-bash: pacman--version: command not found  
cristian@localhost:~> Pacman --version  
-bash: Pacman: command not found  
cristian@localhost:~> grub-intall --version  
-bash: grub-intall: command not found  
cristian@localhost:~> exit  
logout  
  
Welcome to openSUSE MicroOS (x86_64) - Kernel 6.19.2-1-default (tty1).  
enp0s3: 10.0.2.15 fd17:625c:f037:2:ed24:f654:99a:0b82  
  
localhost login: cristian  
Password:  
Last login: Wed Feb 25 14:55:23 -05 2026 on tty1  
cristian@localhost:~> pacman --version  
-bash: pacman: command not found  
cristian@localhost:~> _
```

- **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.

- **Procesos:** El sistema usa systemd como proceso padre (PID 1). Pude ver la lista de procesos activos con ps aux.

```

root      510  0.0  0.0   0  0 ?          I  14:54  0:00 [kworker/2:3-events_freezable_pwr_efficient]
root      520  0.0  0.0   0  0 ?          S  14:54  0:00 [scsi_eh_2]
root      521  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-scsi_tmf_2]
root      522  0.0  0.0   0  0 ?          S  14:54  0:00 [scsi_eh_2h]
root      523  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-scsi_tmf_3]
root      536  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:2h-kblockd]
root      835  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-worker]
root      836  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-dalloc]
root      877  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-falloc]
root      830  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-scan]
root      839  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-fxopl]
root      848  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-endio]
root      841  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-endio-metal]
root      842  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-rmw]
root      843  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-endio-wrie]
root      844  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-freezerace-wrie]
root      845  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-delayed-metal]
root      846  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:3-btrfs-qgroup-rescan]
root      847  0.0  0.0   0  0 ?          S  14:54  0:00 [btrfs-cleaner]
root      848  0.0  0.0   0  0 ?          S  14:54  0:00 [btrfs-trim]
root      900  0.0  0.2 58762 17939 ?  Ss 14:54  0:00 [kworker/16:5-systemd-journald]
root      1624  0.0  0.0   0  0 ?          I  14:54  0:00 [kworker/16:5-btrfs-endio-wrie]
root      1628  0.0  0.2 51972 22208 ?  Ss 14:54  0:00 [usr/lib/systemd/systemd-udevd]
root      1121  0.0  0.0 13996 2128 ?  Sks 14:54  0:00 [usr/sbin/auditd]
message+ 1145  0.0  0.0 5204 4308 ?  Ss 14:54  0:00 [usr/sbin/ibus-broker-launch --scope system --audit]
polkitd   1145  0.0  0.1 386420 8952 ?  Ss 14:54  0:00 [ibus-broker-launch --scope user --audit]
polkitd   1158  0.0  0.0 4936 3164 ?  Ss 14:54  0:00 [usr/libexec/rebootcmd --service]
root      1163  0.0  0.3 32952 2396 ?  Ss 14:54  0:00 [usr/libexec/rebootcmd --service]
root      1168  0.1  0.8 210776 2676 ?  Ss 14:54  0:00 [btrfs-MC-client]
root      1195  0.0  0.2 479972 22912 ?  Ss 14:54  0:00 [usr/sbin/NetworkManager --no-daemon]
root      1200  0.0  0.0 389220 3761 ?  S  14:54  0:00 [usr/bin/ibus-qt5-service --pidfile /var/run/ibus-qt5-service.sh]
root      1253  0.0  0.1 391920 13672 ?  Ss 14:54  0:00 [usr/bin/ibus-qt5-service]
root      1260  0.0  0.0   0  0 ?          I< 14:54  0:00 [kworker/2:2h-kblockd]
chromog  1275  0.0  0.0 88580 6952 ?  Ss 14:54  0:00 [usr/sbin/chronog -n]
root      1345  0.0  0.0 12668 7756 ?  Ss 14:54  0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startup
root      1820  0.2  0.0   0  0 ?          I  15:00  0:00 [kworker/3:0-events]
root      1821  0.0  0.0   0  0 ?          I  15:00  0:00 [kworker/4:0-events]
root      1845  0.1  0.0   0  0 ?          I  15:00  0:00 [kworker/1:0-events]
root      1889  0.0  0.0 9949 6768 ?  Ss 15:01  0:00 login -- cristian
root      1893  0.1  0.0   0  0 ?          I  15:01  0:00 [kworker/2:0-events]
root      1906  0.0  0.0   0  0 ?          I< 15:01  0:00 [kworker/2:0h-kblockd]
cristian 1334  0.1  0.3 35744 25716 ?  Ss 15:02  0:00 [usr/lib/systemd/systemd --user]
cristian 1335  0.0  0.0 23040 4224 ?  S  15:02  0:00 [sd-pam]
cristian 1946  0.0  0.0 7749 6444 ttys1  Ss 15:02  0:00
root      2821  0.0  0.0   0  0 ?          I  15:05  0:00 [kworker/0:1-events]
root      2822  0.0  0.0   0  0 ?          I  15:05  0:00 [kworker/1:1-cgroup_free]
root      2844  0.1  0.0 5732 4436 ttys0  Ss+ 15:07  0:00 /sbin/getty --noreset --noclear --keep-baud 115200,57600,38400,9600 - vt220
cristian 2846  0.0  0.0 9524 6280 ttys1  R+ 15:07  0:00 ps aux
cristian@localhost:~>
```

- **Memoria:** El kernel asigna la RAM dinámicamente. Verifiqué el consumo de la máquina virtual con free -h.

```

cristian@localhost:~> free -h
              total        used        free      shared  buff/cache   available
Mem:       7.6Gi      441Mi      7.16i     1.0Mi      204Mi      7.16i
Swap:            0B          0B          0B
cristian@localhost:~> _
```

- **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.

```

cristian@localhost:~> df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb3       29G   943M  19G  5% /
/dev/sdb3       29G   943M  19G  5% /etc
/dev/sdb3       29G   943M  19G  5% /root
/dev/sdb3       29G   943M  19G  5% /var
devtmpfs        3.8G    0  3.8G  0% /dev
tmpfs          3.8G    0  3.8G  0% /dev/shm
efivarsfs      256K  37K  215K 15% /sys/firmware/efi/efivars
tmpfs          1.6G  1.1M  1.6G  1% /run
tmpfs          3.8G    0  3.8G  0% /tmp
tmpfs          1.0M    0  1.0M  0% /run/credentials/systemd-journal.service
/dev/sdb3       29G   943M  19G  5% /.snapshots
/dev/sdb3       29G   943M  19G  5% /boot/grub2/i386-pc
/dev/sdb3       29G   943M  19G  5% /boot/writable
/dev/sdb3       29G   943M  19G  5% /home
/dev/sdb2       28M   4.2M  18M  21% /boot/efi
/dev/sdb3       29G   943M  19G  5% /opt
/dev/sdb3       29G   943M  19G  5% /boot/grub2/x86_64-efi
/dev/sdb3       29G   943M  19G  5% /srv
/dev/sdb3       29G   943M  19G  5% /usr/local
tmpfs          1.0M    0  1.0M  0% /run/credentials/getty@tty1.service
tmpfs          774M  8.8K  774M  1% /run/user/1000
tmpfs          1.0M    0  1.0M  0% /run/credentials/serial-getty@ttyS0.service
cristian@localhost:~> sudo snapper list
[sudo] password for root:
#  Type | Pre # | Date           | User | Used Space | Cleanup | Description | Userdata
#  single |       | Wed 18 Feb 2026 04:43:13 PM -05 | root | 48.00 KIB |         | current
1* single |       | Wed 25 Feb 2026 09:47:02 AM -05 | root | 48.00 KIB |         | first root filesystem
2 single |       | Wed 25 Feb 2026 18:00:06 AM -05 | root | 48.00 KIB | timeline | After jeos-firstboot configuration | important=yes
3 single |       | Wed 25 Feb 2026 11:00:00 AM -05 | root | 48.00 KIB | timeline |
4 single |       | Wed 25 Feb 2026 12:00:02 PM -05 | root | 48.00 KIB | timeline |
5 single |       | Wed 25 Feb 2026 01:00:06 PM -05 | root | 48.00 KIB | timeline |
6 single |       | Wed 25 Feb 2026 03:00:05 PM -05 | root | 48.00 KIB | timeline |
7 single |       | Wed 25 Feb 2026 03:00:05 PM -05 | root | 48.00 KIB | timeline |
cristian@localhost:~> -

```

- **Dispositivos (E/S):** El kernel administra el hardware conectado. Listé mis discos virtuales con `lsblk`.

```

cristian@localhost:~> lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda    8:0    0   8G  0 disk
sdb    8:16   0  29G  0 disk
|---sdb1 8:17   0   2M  0 part
|---sdb2 8:18   0  28M  0 part /boot/efi
|---sdb3 8:19   0  29G  0 part /usr/local
                           /srv
                           /boot/grub2/x86_64-efi
                           /opt
                           /home
                           /boot/writable
                           /boot/grub2/i386-pc
                           /.snapshots
                           /var
                           /root
                           /etc
                           /
sr0   11:0    1 1024M  0 rom
cristian@localhost:~>

```

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, pasárle 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.

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.