

# Arquitectura y sistemas operativos

Trabajo práctico integrador: Virtualización.

Profesor: Emmanuel Avellaneda.

Alumnos: Luis Mariano Rivera Ortiz, Gustavo Hernan Campestre.

Fecha de Entrega: 3/10/2025.

## Introducción

El tema de la **virtualización** fue elegido porque constituye una de las tecnologías más relevantes en el ámbito actual de la informática, permitiendo crear entornos de prueba y producción sin necesidad de contar con múltiples equipos físicos. Esta característica la convierte en una herramienta fundamental tanto para el aprendizaje como para el trabajo profesional, ya que posibilita la instalación y configuración de distintos sistemas operativos y servicios de manera flexible y segura. Asimismo, la virtualización es utilizada de forma cotidiana en empresas e instituciones, lo que refuerza la pertinencia de abordarla en un trabajo práctico de formación técnica.

En la formación de un **técnico en programación**, la virtualización ocupa un lugar central, dado que permite comprender de manera práctica la interacción entre hardware, sistemas operativos y aplicaciones. El estudiante, a través de estas prácticas, adquiere habilidades para gestionar entornos virtuales que le servirán para implementar, probar y depurar software en distintos escenarios. Además, la virtualización le ofrece la posibilidad de reproducir problemas, realizar simulaciones y experimentar con configuraciones avanzadas, sin afectar el funcionamiento del equipo principal ni requerir recursos adicionales. De esta manera, se fomenta la autonomía, la capacidad de resolución de problemas y la preparación para entornos de desarrollo profesionales.

El trabajo se propone como objetivos principales: afianzar los conocimientos teóricos sobre los conceptos de hipervisores, redes virtuales y snapshots; desarrollar destrezas en la instalación y configuración de máquinas virtuales con sistemas operativos de uso habitual en la industria; y validar el funcionamiento de servicios básicos, como la instalación de un servidor web, en un entorno controlado. A través de estos objetivos, el estudiante no solo refuerza su formación técnica, sino que también incorpora una metodología de trabajo que le permitirá desempeñarse con mayor eficiencia y seguridad en su futura práctica profesional.

**Sandbox:**

Un sandbox es un entorno aislado que permite ejecutar aplicaciones o código de forma segura sin afectar al sistema principal. Se utiliza para probar software desconocido, analizar virus o experimentar sin riesgos. Algunos ejemplos son Windows Sandbox, Sandboxie, Firejail en Linux o el aislamiento de pestañas en navegadores como Chrome y Firefox. También existen sandboxes en entornos de programación, como los playgrounds de Python, Node.js o Google Colab.

En cambio, VirtualBox, VMware e Hyper-V no son sandboxes sino programas de virtualización que crean máquinas virtuales completas, con su propio sistema operativo, hardware virtual y recursos asignados.

La diferencia principal es que una máquina virtual es como tener otra computadora completa dentro de la tuya, mientras que una sandbox es como un cuarto aislado en la misma casa donde podés probar cosas sin que se propaguen al resto.

**Marco teórico**

Virtualización de hardware: Máquina Virtual Linux.

Hipervisores: Virtual Box.

Imágenes ISO: ubuntu-24.04.3-live-server-amd64.iso.

Redes virtuales: Puente (Bridge).

Putty: Cliente SSH.

Python3: Lenguaje de programación.

GitHub: Repositorio para guardar, versionar y colaborar en proyectos de software.

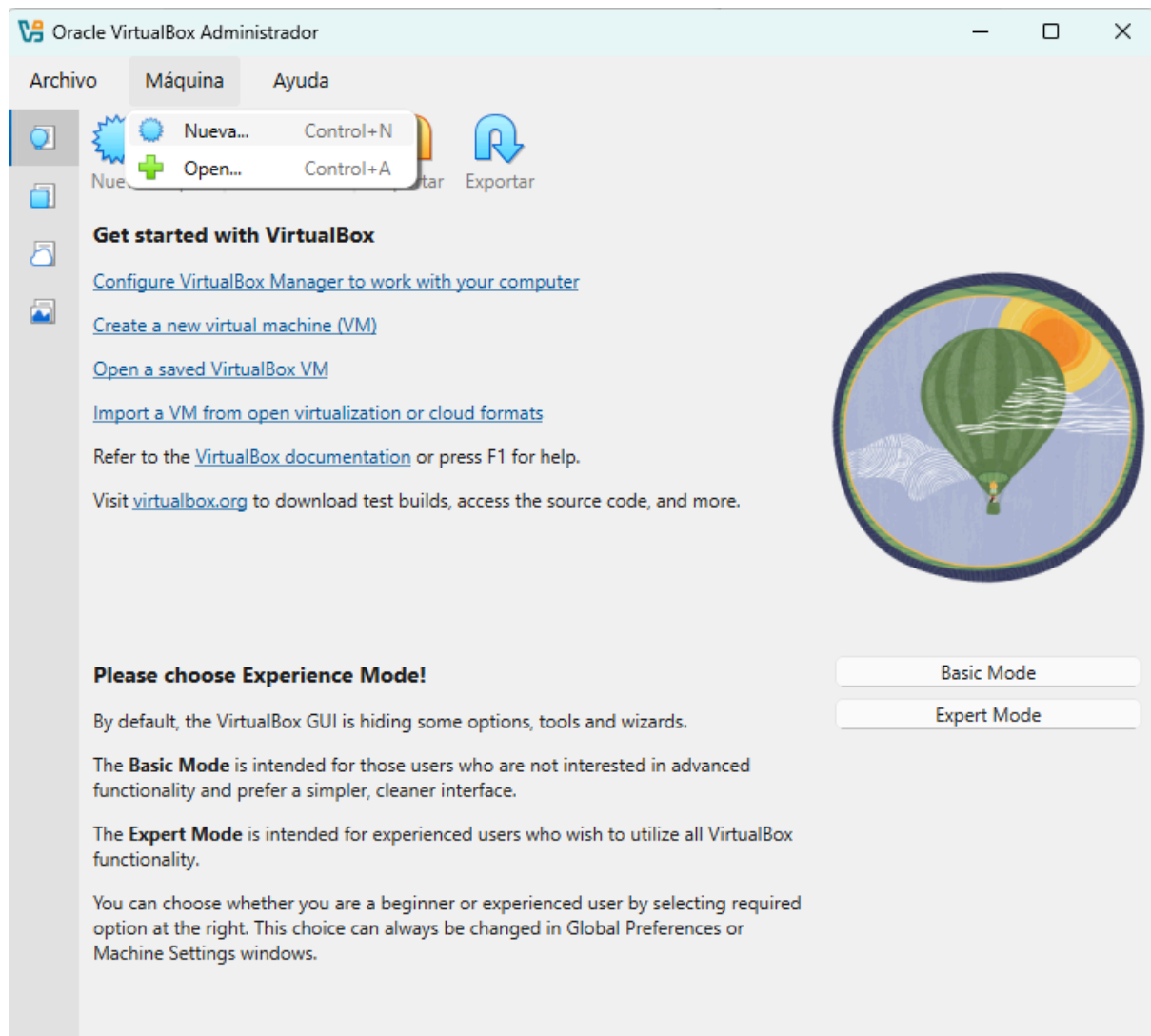
## Caso práctico

-Descargamos Virtual Box de la web <https://www.virtualbox.org/wiki/Downloads> . VirtualBox es un software de virtualización gratuito que permite crear y ejecutar máquinas virtuales para usar varios sistemas operativos en una misma computadora.

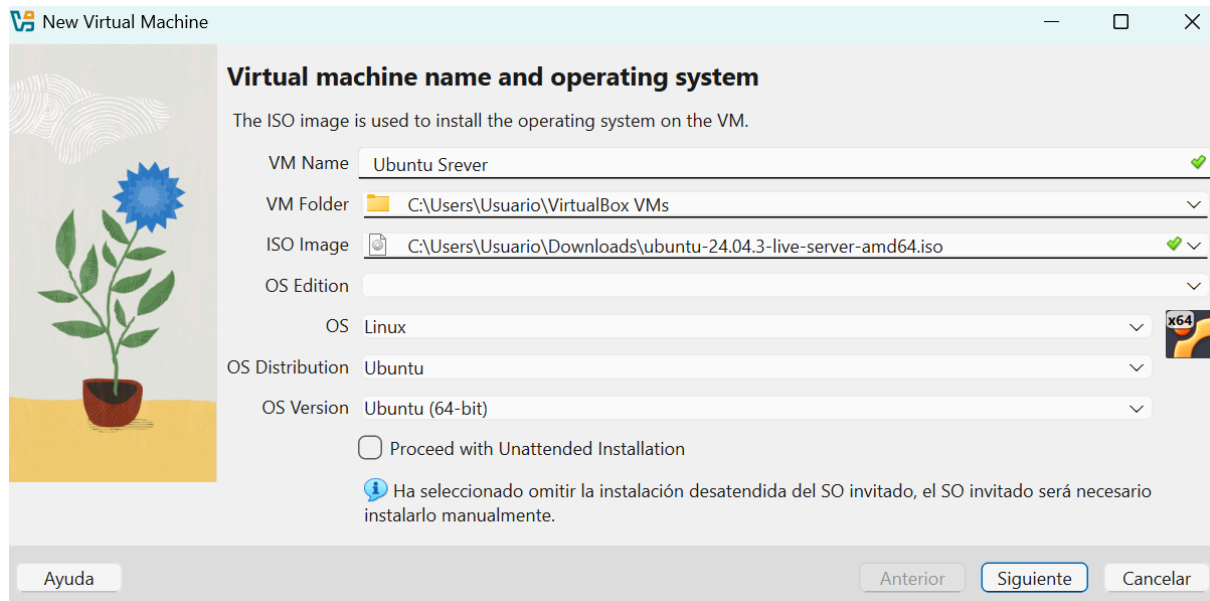
-Descargamos la Imagen ISO Ubuntu 24.04.3 LTS de la web oficial <https://ubuntu.com/download/server>.

Una **ISO de Ubuntu** es un archivo que contiene una copia exacta del instalador de este sistema operativo. Se usa para crear un USB o DVD booteable y así instalar o probar Ubuntu en una computadora, o también para montar en un software de **virtualización** (como VirtualBox o VMware) y ejecutar Ubuntu sin necesidad de grabarlo en un medio físico.

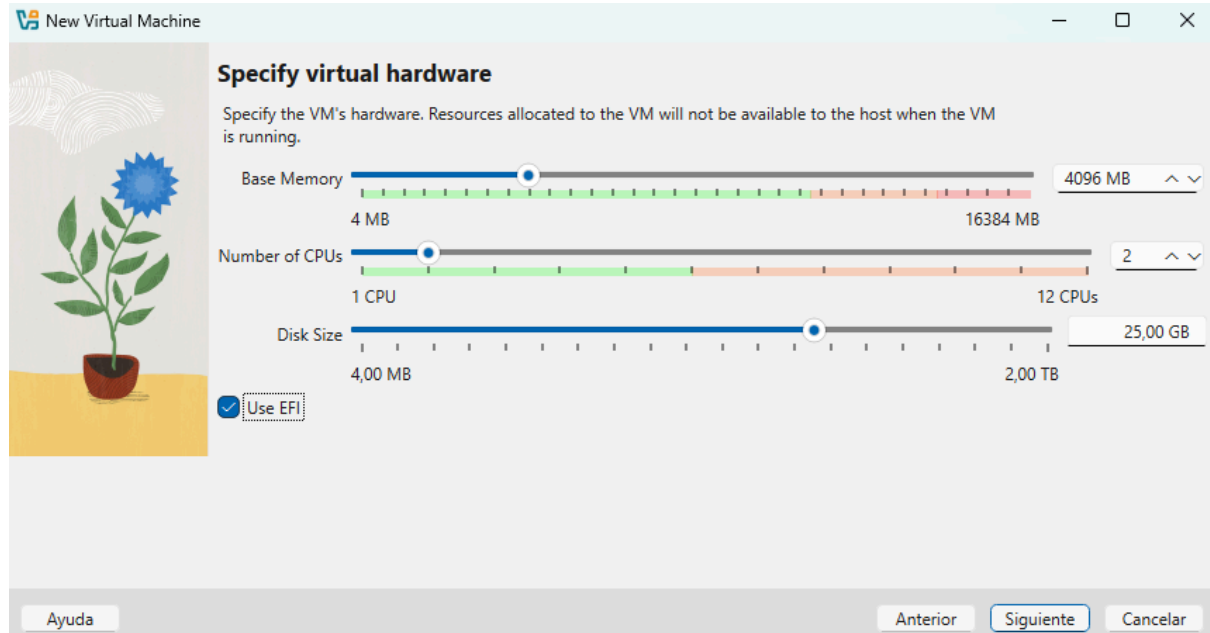
-Abrimos el programa Virtual box y vamos a la pestaña maquina→nueva.



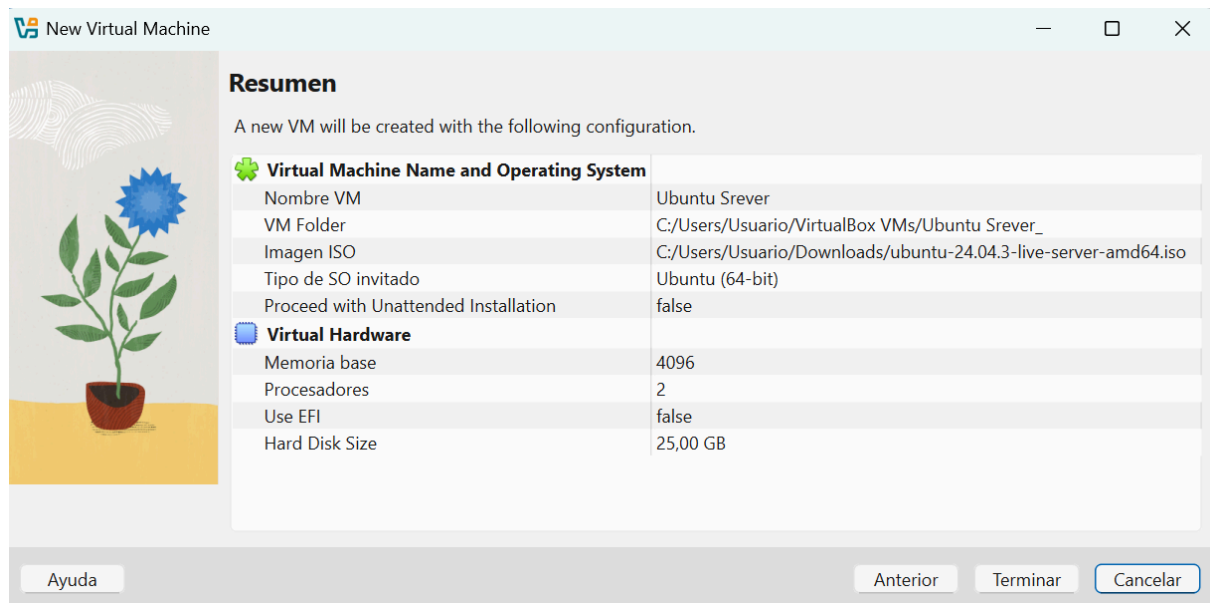
-Se define un nombre para la VM y se selecciona el directorio en donde se descargó la imagen ISO y se pone siguiente.



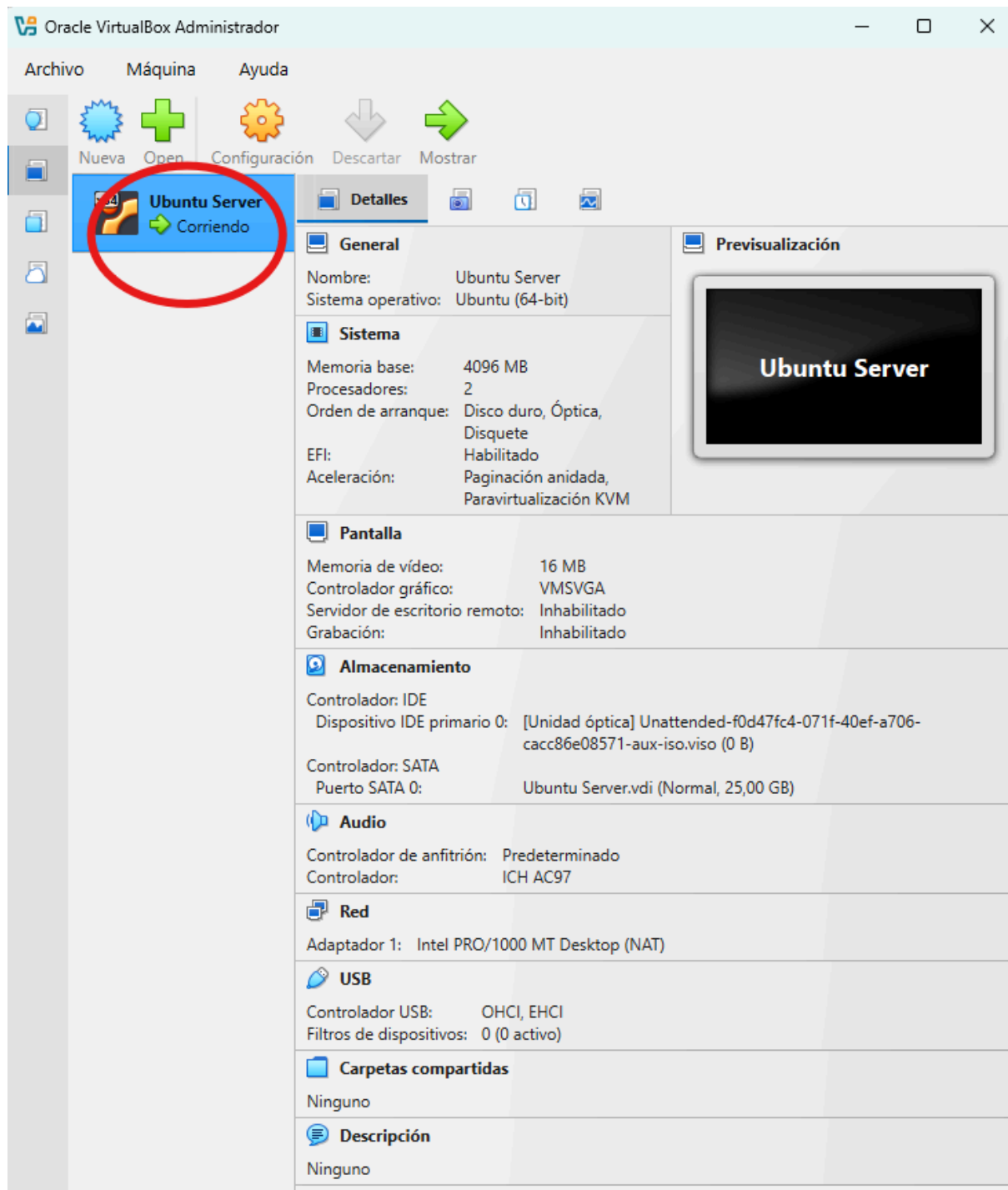
-Se definen los recursos deseados. En nuestro caso definimos 4 gigas de memoria RAM, 2 Cores de CPU y 25 Gigas de espacio en disco.



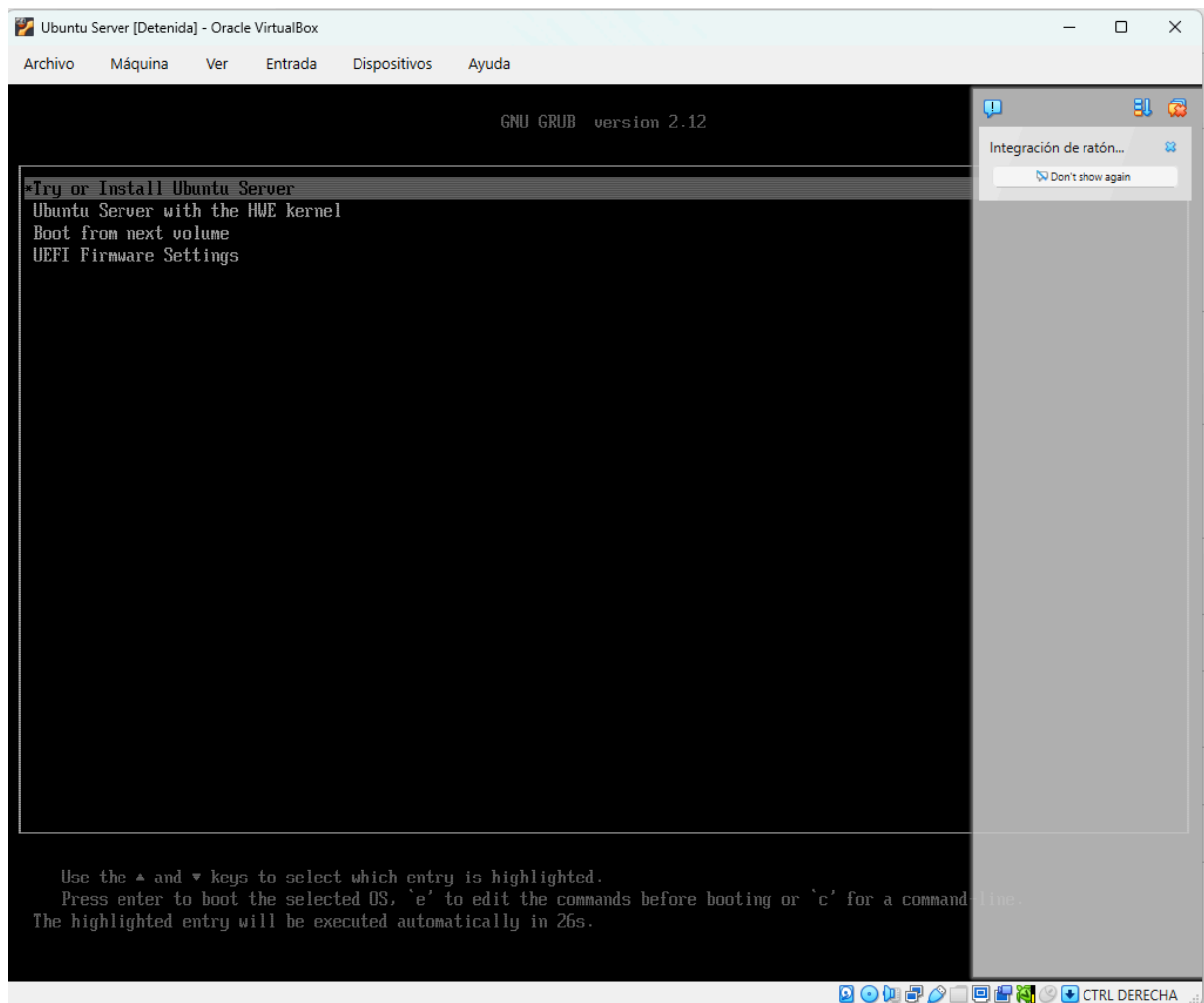
-Se muestra un resumen de las opciones elegidas y se presiona terminar para iniciar el proceso de instalación.



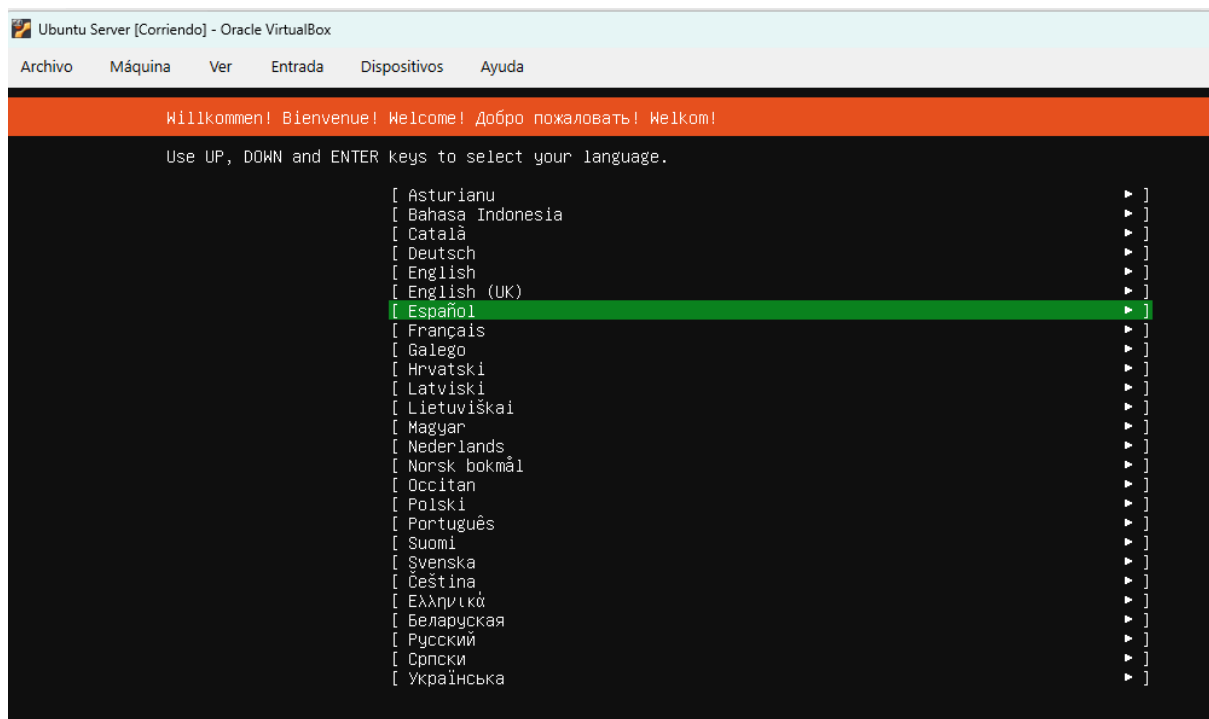
-Maquina virtual creada. Para acceder se hace click en el sector indicado en la imagen.



-Se inicia el proceso de instalación eligiendo desde el gestor de arranque de Linux (GRUB) la primera opción. GRUB (GRand Unified Bootloader) es un gestor de arranque que permite elegir y cargar sistemas operativos al iniciar la computadora

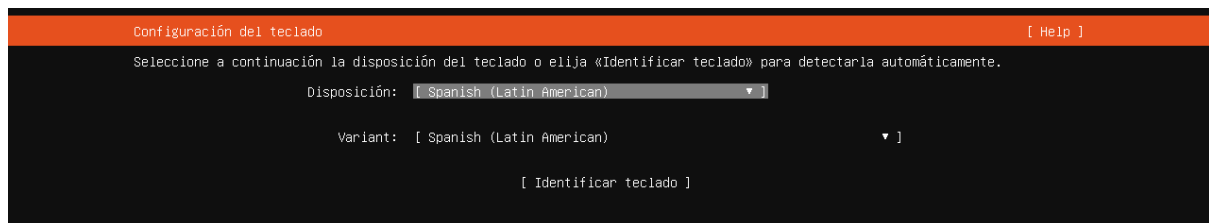


-Se elige el idioma

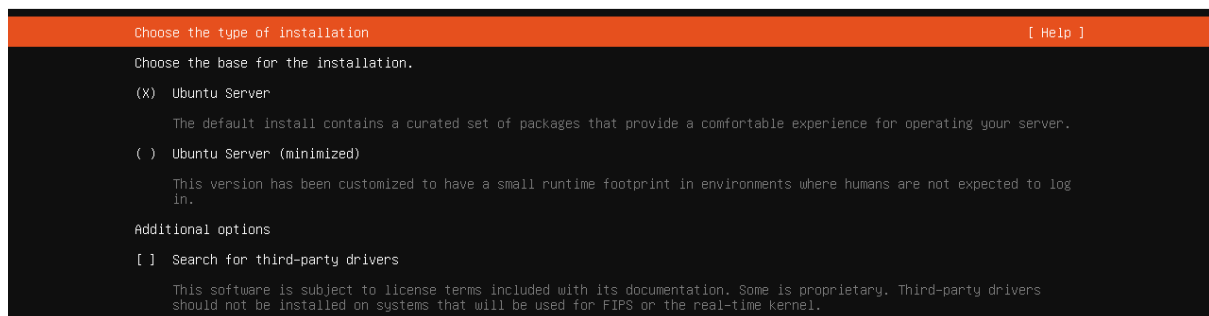




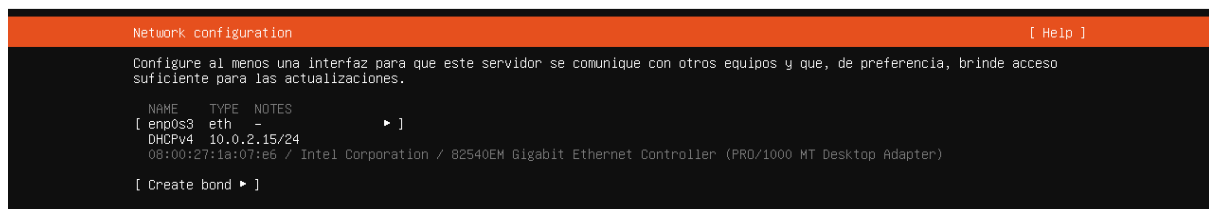
-Se define la configuración del teclado



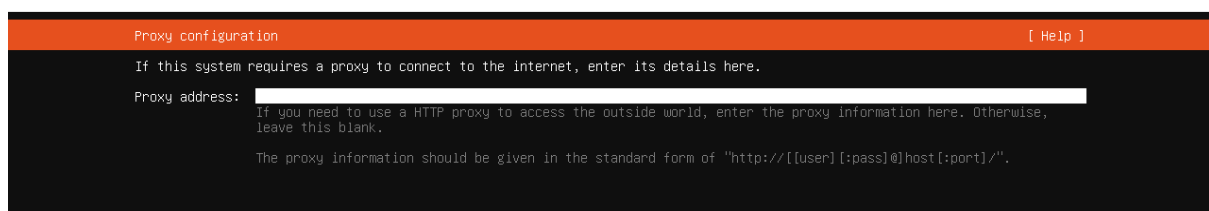
-Tipo de instalación. Dejamos lo que está por defecto.



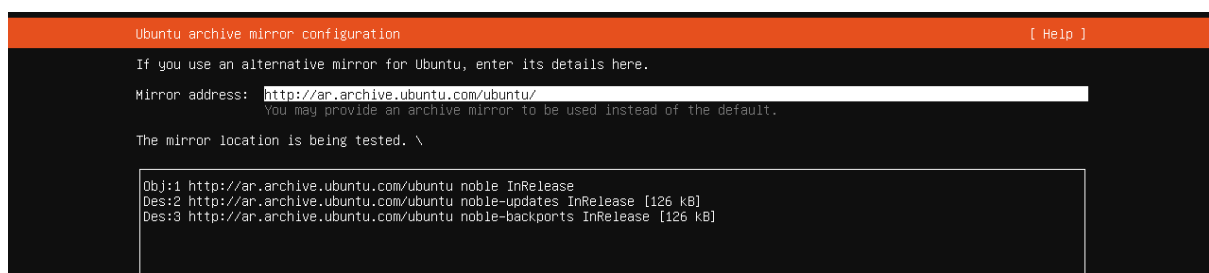
-Configuración de red. Dejamos lo que está por defecto.



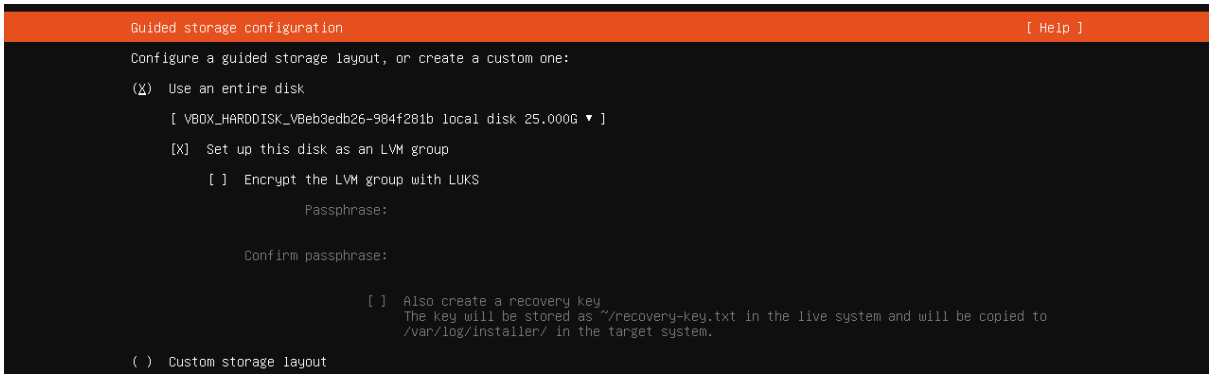
-Proxy. No usamos un servidor proxy para salir a Internet. Dejamos lo que está por defecto.



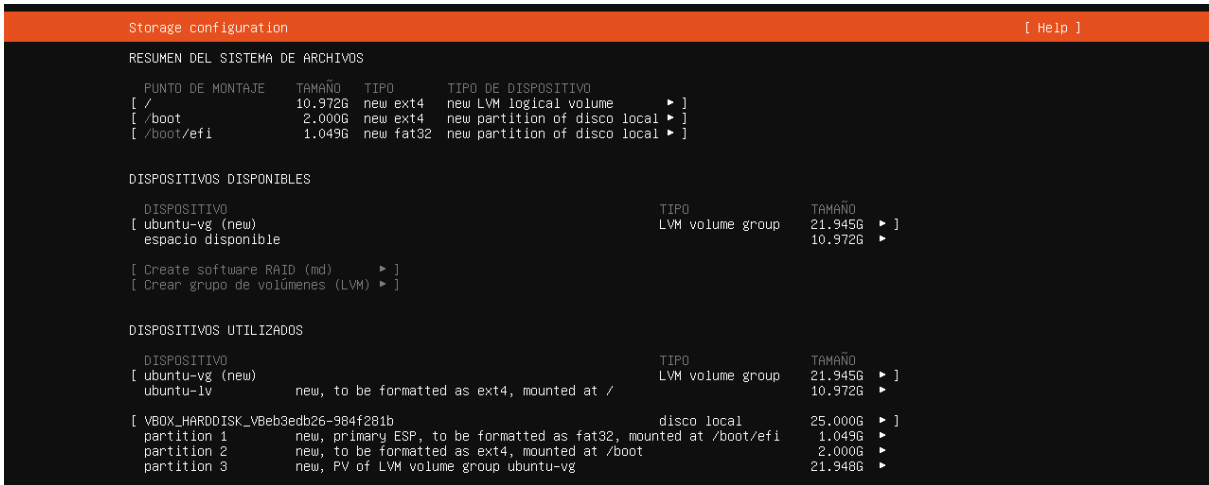
-Lee lista de paquetes.



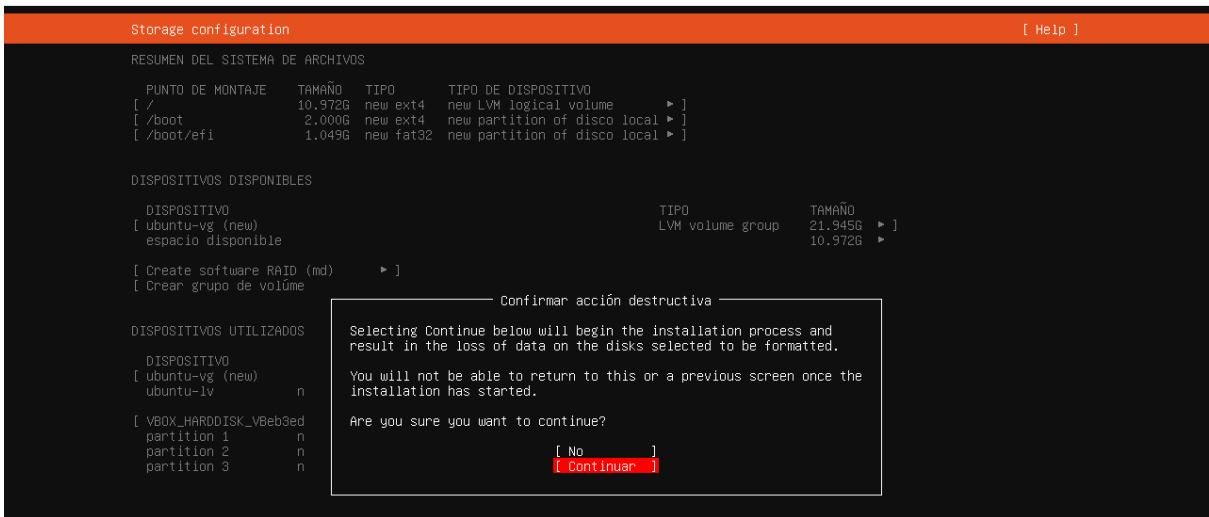
-Define que va a crear un disco nuevo. Dejamos lo que está por defecto.



-Muestra como queda el sistema de archivos. Dejamos lo que está por defecto.



-Pide confirmación por seguridad para proceder con la instalación.



-Pide definir usuario y clave.

Profile configuration [ Help ]

Enter the username and password you will use to log in to the system. You can configure SSH access on a later screen, but a password is still needed for sudo.

Su nombre:

Your servers name:   
The name it uses when it talks to other computers.

Elija un nombre de usuario:

Elija una contraseña:

Confirme la contraseña:

-Saltamos esta opción de actualización. Dejamos lo que está por defecto.

Upgrade to Ubuntu Pro [ Help ]

Upgrade this machine to Ubuntu Pro for security updates on a much wider range of packages, until 2034. Assists with FedRAMP, FIPS, STIG, HIPAA and other compliance or hardening requirements.

[ About Ubuntu Pro ► ]

( ) Enable Ubuntu Pro

(X) Skip for now

You can always enable Ubuntu Pro later using the 'pro attach' command.

-Pregunta si queremos instalar el servicio SSH. Dejamos lo que está por defecto.

SSH configuration [ Help ]

You can choose to install the OpenSSH server package to enable secure remote access to your server.

(X) Instalar servidor OpenSSH

[X] Permitir autenticación con contraseña por SSH

[ Import SSH key ► ]

AUTHORIZED KEYS

No authorized key

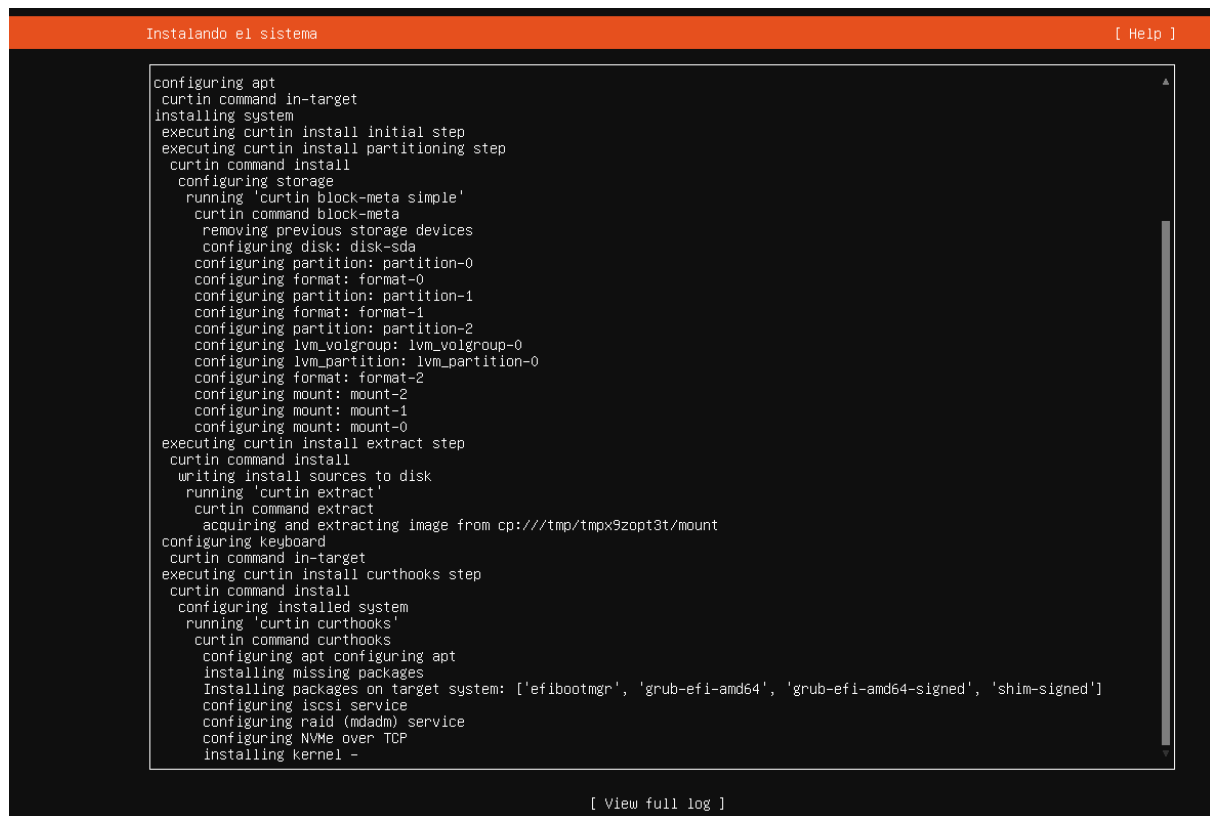
-No seleccionamos opciones. Dejamos lo que está por defecto.

Featured server snaps [ Help ]

These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see more details of the package, publisher and versions available.

[ ] microk8s	canonical✓	Kubernetes for workstations and appliances
[ ] nextcloud	nextcloud✓	Nextcloud Server - A safe home for all your data
[ ] wekan	xet7	Open-Source kanban
[ ] kata-containers	katacontainers✓	Build lightweight VMs that seamlessly plug into the containers ecosystem
[ ] docker	canonical✓	Docker container runtime
[ ] canonical-livpatch	canonical✓	Canonical Livepatch Client
[ ] rocketchat-server	rocketchat✓	Rocket.Chat server
[ ] mosquito	mosquitto✓	Eclipse Mosquitto MQTT broker
[ ] etcd	canonical✓	Resilient key-value store by CoreOS
[ ] powershell	canonical✓	PowerShell for every system!
[ ] sabnzbd	safihre	SABnzbd
[ ] wormhole	snappcrafters	get things from one computer to another, safely
[ ] aws-cli	aws✓	Universal Command Line Interface for Amazon Web Services
[ ] google-cloud-sdk	google-cloud-sdk✓	Google Cloud SDK
[ ] scli	softlayer	Python based SoftLayer API Tool.
[ ] doctl	digitalocean✓	The official DigitalOcean command line interface
[ ] postgresql10	cmd✓	PostgreSQL is a powerful, open source object-relational database system.
[ ] keepalived	keepalived-project✓	High availability VRRP/BFD and load-balancing for Linux
[ ] prometheus	canonical✓	The Prometheus monitoring system and time series database
[ ] lxd	canonical✓	LXD - container and VM manager

-Comienzo de la instalación.



```
Instalando el sistema [ Help ]

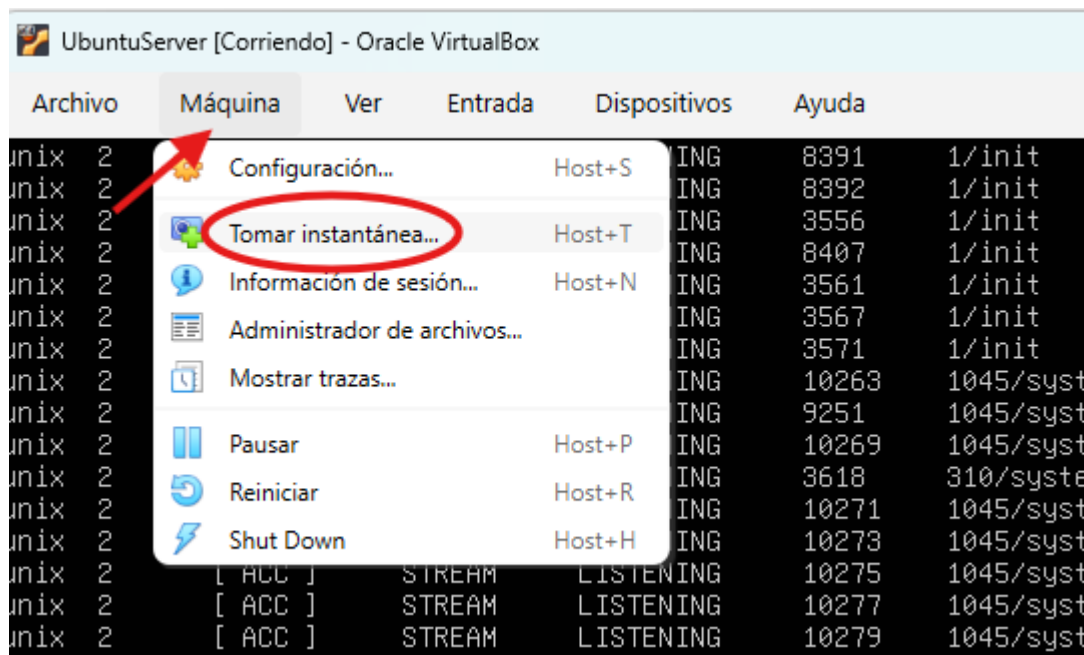
configuring apt
curtin command in-target
installing system
executing curtin install initial step
executing curtin install partitioning step
curtin command install
  configuring storage
    running 'curtin block-meta simple'
    curtin command block-meta
      removing previous storage devices
      configuring disk: disk-sda
      configuring partition: partition-0
      configuring format: format-0
      configuring partition: partition-1
      configuring format: format-1
      configuring partition: partition-2
      configuring lvm_volgroup: lvm_volgroup-0
      configuring lvm_partition: lvm_partition-0
      configuring format: format-2
      configuring mount: mount-2
      configuring mount: mount-1
      configuring mount: mount-0
  executing curtin install extract step
  curtin command install
    writing install sources to disk
    running 'curtin extract'
    curtin command extract
      acquiring and extracting image from cp:///tmp/tmpx9zopt3t/mount
  configuring keyboard
  curtin command in-target
executing curtin install curthooks step
curtin command install
  configuring installed system
  running 'curtin curthooks'
  curtin command curthooks
    configuring apt configuring apt
    installing missing packages
    Installing packages on target system: ['efibootmgr', 'grub-efi-amd64', 'grub-efi-amd64-signed', 'shim-signed']
    configuring iscsi service
    configuring raid (mdadm) service
    configuring NVMe over TCP
    installing kernel -
```

[ View full log ]

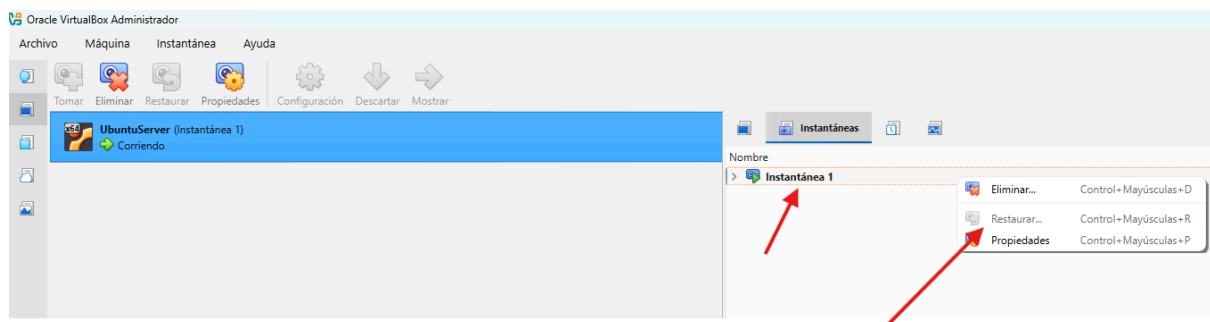
-Una vez iniciada la VM se coloca el usuario y clave definido en la instalación para acceder a la consola de administración.

-Si queremos guardar un estado actual de la Máquina virtual podemos crear un *snapshot* . Un *snapshot* es una captura del estado actual de una máquina virtual en un instante determinado. Permite conservar la configuración, la memoria y el contenido del disco, posibilitando restaurar la VM a ese punto en caso de fallas, errores en la configuración o pruebas de software.

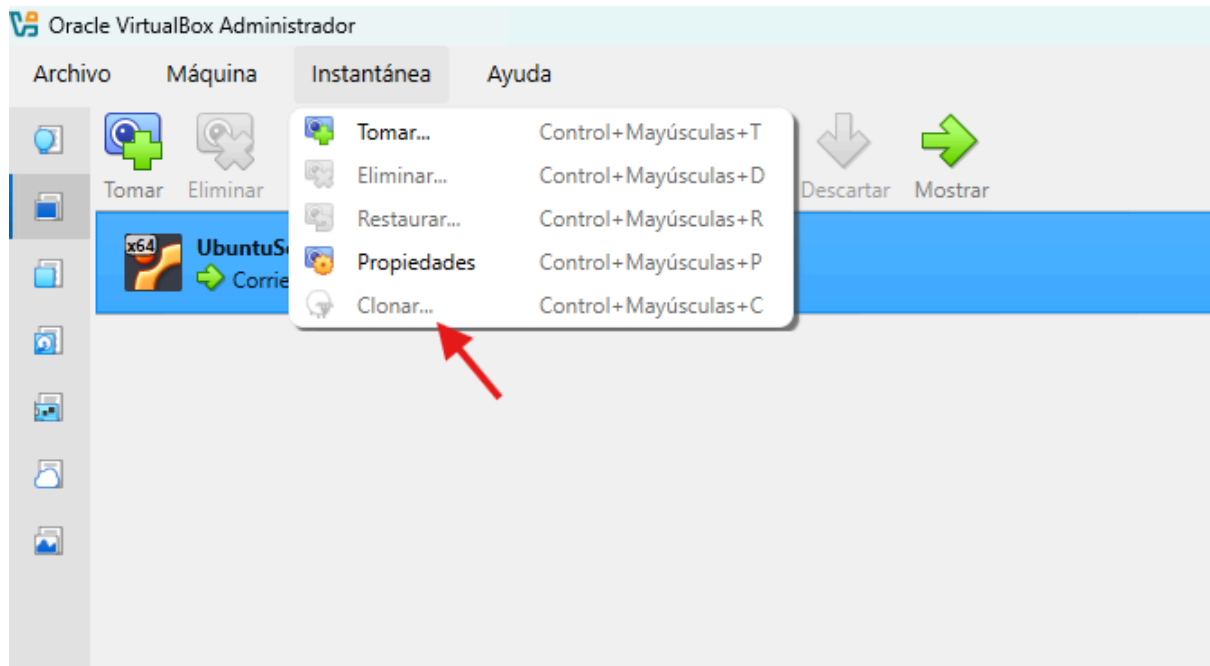
Le definimos un nombre y una fecha para saber cuando fue tomada la instantánea.



-En el caso que queramos volver atrás el estado de una VM debemos ir al panel principal, seleccionar la VM y elegir la fecha del snapshot que queremos.



-Otra opción es crear un *clon* de la máquina virtual. Un clon es una copia completa e independiente de la VM original, que se puede ejecutar por separado y utilizar como si fuera una nueva máquina virtual, sin afectar a la original.



-Continuando con la configuración de la máquina virtual:

-Se ejecuta el comando `sudo su` para tener administración de superusuario(root).  
Permite ejecutar comandos con privilegios de administrador.

-Se chequea que la interfaz de red tenga IP con el comando `ip addr`.

-Se chequea el acceso a internet con el comando `ping 8.8.8.8`.

-Se chequea la resolución DNS con el comando `nslookup google.com`.

-Se instala el servicio **SSH** con el comando `apt-get install ssh`.

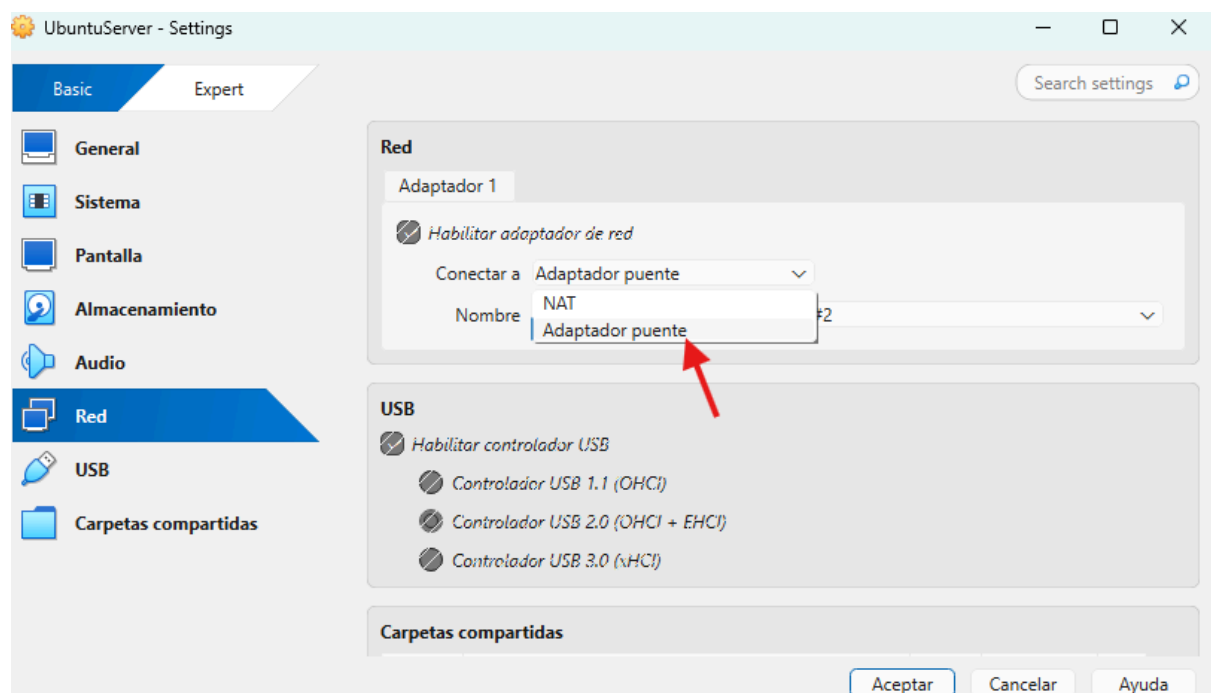
-Se habilita el firewall para las conexiones hacia el puerto 22 con el comando `ufw allow 22/tcp`, antes de activar el firewall.

-Se activa el firewall con `sudo ufw enable`.

-Se chequea que el servicio esté corriendo con `systemctl status ssh`.

-Se instala el paquete **net-tools** con el comando `apt-get install net-tools`. Este incluye, entre otras herramientas, el comando `netstat` para ver si los puertos están levantados. En nuestro caso queremos ver que el puerto 22 del servicio SSH este activo.

-Se cambia la configuración de la placa de red que está en modo **NAT (Network Address Translation)** a modo **Bridge (Puente)** para que tenga acceso directo a la red como una PC conectada físicamente. El modo NAT crea una red virtual interna en VirtualBox que para salir hacia afuera lo hace a través de la IP de la placa física. El objetivo es tener visibilidad directa y poder conectarnos con un cliente SSH, con el fin de hacer la administración más dinámica y fluida.



-Se chequea la nueva IP con el comando `ip addr` o `ipconfig`. Debe ser del rango de la placa física.

-Se utiliza el cliente de SSH **PuTTY** en la PC física para conectarnos a la IP nueva de la VM. En dicho cliente se debe definir también el puerto 22.

Todo el escenario anteriormente mencionado confluye en un entorno que se asemeja mucho más a una situación real de producción. De la misma forma, en un entorno profesional se utilizan plataformas como **vCenter de VMware** o **Hyper-V de**

**Microsoft**, que permiten administrar múltiples servidores virtuales de manera centralizada, con redes virtuales avanzadas, alta disponibilidad y mayor seguridad. .

## **GIT**

- Comenzamos iniciando sesión en <https://github.com>
- Se crea el repositorio desde el botón “New”
- Se define el nombre, una descripción , la visibilidad (Público/Privado) en nuestro caso va a ser Público y se tilda la opción Readme.
- Creo una carpeta con el proyecto en mi máquina con el comando “git clone <https://github.com/ghernanc/TrabajoPracticoIntegradorAySO.git>”. El comando se introduce en el home del usuario, para que se cree allí.

## **Linux**

- Se instala la última versión de python con el comando `sudo apt install python3 -y`
- Se instala la última versión de git con el comando `sudo apt install git -y`
- Se crea una carpeta con el comando “`mkdir python`” en el home del usuario. que en nuestro caso es `/home/usuario/python`.
- Se crea un archivo [programa.py](#) y se guarda en la carpeta antes mencionada.
- Se crea un programa de prueba llamado [progama.py](#).

### **Proceso para subir el [archivo.py](#) en el repositorio:**

- Abrimos la terminal y nos ubicamos dentro de la carpeta del proyecto:



```
cd /home/usuario/python
```

- Inicializamos Git en la carpeta (solo la primera vez):

```
git init (Esto crea un repositorio local en tu carpeta.)
```

- Vinculamos el repositorio local con el repositorio remoto en GitHub:

```
git remote add origin  
https://github.com/ghernanc/TrabajoPracticoIntegradorAyS0.git
```

- Configuramos nuestro usuario y correo para Git :

```
git config --global user.name "Luis-Rivera90"
```

```
git config --global user.email "luis.mariano590@gmail.com"
```

- Agregamos el archivo [programa.py](#) al área de staging:

```
git add programa.py (Esto indica que solo queremos incluir ese archivo  
en el próximo commit.)
```

- Antes de hacer commit verificamos el estado de los archivos, si estan en verde,  
están preparados para el commit

```
git status
```

- Creamos un commit con un mensaje descriptivo:

```
git commit -m "Primer commit: agrego programa.py"
```

- Nos aseguramos de que la rama principal se llame **main**:

```
git branch -M main
```

- Hacemos el primer push al repositorio remoto:

```
git push -u origin main
```

Git pedirá:

- **Username:** nuestro usuario de GitHub
- **Password:** el Personal Access Token (Classic) generado previamente con permisos en repo.

- Para subir nuevos cambios usamos:

```
git add programa.py
```

```
git commit -m "Descripción de los cambios"
```

```
git push
```

## Metodología Utilizada

- Instalación de VirtualBox en Windows mediante el instalador **.exe**.
- Descarga de la imagen ISO oficial de Ubuntu Server 24.04.3 LTS desde la página oficial.
- Creación de la máquina virtual en VirtualBox, definiendo recursos de hardware: 4 GB de RAM, 2 núcleos de CPU y 25 GB de disco.
- Configuración inicial de instalación de Ubuntu: idioma, teclado, usuario y clave.
- Configuración de red en modo Bridge para otorgar acceso externo a la VM como si fuera una máquina física.
- Instalación de servicios básicos en la VM: SSH, net-tools, firewall (ufw).

- Verificación de conectividad mediante `ip addr`, `ping` y `nslookup`.
- Uso del cliente PuTTY en el host para acceder a la VM vía SSH.
- Creación de snapshots para guardar estados de la VM y posibilidad de restaurarlos.
- Creación de un clon de la VM como copia independiente para pruebas adicionales.
- Subida de un archivo Python de prueba a un repositorio de GitHub utilizando comandos de Git.

## Resultados Obtenidos

- El sistema operativo invitado Ubuntu se instaló y ejecutó correctamente en VirtualBox.
- El acceso remoto mediante SSH permitió administrar la máquina de forma más práctica.
- El servidor virtual logró conectarse con el repositorio en GitHub, confirmando la integración con servicios externos.
- Se comprobó el correcto funcionamiento de la configuración de red en modo Bridge.
- Se verificó la utilidad de snapshots y clones para experimentar sin riesgo de perder configuraciones.
- Se comprendió la interacción entre el sistema host y el invitado en un entorno de virtualización.

---

## Conclusiones

- La virtualización simplifica la creación de entornos de prueba y aprendizaje sin necesidad de hardware adicional.

- VirtualBox se presenta como una herramienta accesible, gratuita y potente para gestionar máquinas virtuales.
- Los snapshots y clones demostraron ser mecanismos útiles para mantener versiones seguras de la VM y para realizar pruebas sin comprometer el sistema principal.
- La práctica permitió afianzar conceptos clave de redes, administración de sistemas Linux y uso de herramientas de colaboración como GitHub.
- Este tipo de entornos favorece la autonomía del estudiante y lo prepara para escenarios profesionales donde se emplean hipervisores más avanzados como VMware vCenter o Microsoft Hyper-V.

## **Anexo:**

Sistema Operativo Host



Windows



ORACLE VM

VirtualBox



ubuntu

Entorno virtualizado en un Windows host, utilizando VirtualBox para correr una VM con Ubuntu

## **Bibliografía**

- Documentación de VirtualBox: <https://www.virtualbox.org/manual/>
- Ubuntu Server Guide: <https://ubuntu.com/server/docs>
- Redes virtuales con VBox: <https://wiki.archlinux.org/title/VirtualBox>
- Github: <https://docs.github.com/es>