

Trabajo Integrador

Virtualización

ANTES DE LA VIRTUALIZACIÓN

DESPUÉS DE LA VIRTUALIZACIÓN

Estudiantes

- Domenicale Doré, Maria Luz - Comisión 4

maria.domenicale@tupad.utn.edu.ar

- Egea Ruiz, Magaly - Comisión 4

magaly.egea@tupad.utn.edu.ar

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Arquitectura y Sistemas Operativos

Docente Titular

Prof. Ariel Enferrel

Docente Tutor

Comisión 4: Nicolás Carcaño

22 de octubre de 2025

Índice

Índice	1
1. Introducción	2
2. Marco Teórico	3
¿Qué es la virtualización?	3
¿Cómo se logra?	3
¿Qué tiene una máquina virtual?	4
¿Qué es un sandbox?	4
Ventajas de la virtualización	4
Ejemplo aplicado	5
3. Caso Práctico	5
Creación del espacio virtual: instalación del sistema y selección de herramientas	5
Inicio de Sesión en Ubuntu Server	19
Arranque del Sistema e instalación de Herramientas	20
Instalación de Python	24
4. Metodología Utilizada	26
Investigación previa (fuentes utilizadas).	26
Etapas de diseño y prueba del código.	26
Herramientas y recursos utilizados (IDE, librerías, control de versiones, etc.).	26
Trabajo colaborativo (reparto de tareas en el grupo de trabajo).	27
5. Resultados Obtenidos	27
6. Conclusiones	28
7. Bibliografía	29
8. Anexos	30
Código Fuente en Python para el Script:	30

1. Introducción

La **virtualización** es una tecnología clave en el campo de la arquitectura de computadoras y los sistemas operativos. Su capacidad para crear entornos aislados y replicables permite a los estudiantes experimentar, aprender y desarrollar habilidades técnicas sin comprometer el sistema principal.

Este trabajo fue elegido por su relevancia en la formación como técnica en programación, ya que permite comprender cómo se gestionan los recursos físicos mediante software, y cómo se pueden simular sistemas completos dentro de una máquina virtual.

El **objetivo principal** es aplicar los conceptos teóricos de virtualización mediante la creación de una máquina virtual con Ubuntu Server, la instalación de Python y la ejecución de un programa en un entorno controlado. A través de este proceso, se busca validar el funcionamiento de la tecnología, reflexionar sobre sus ventajas y documentar el aprendizaje obtenido.

Se eligió **VirtualBox** como herramienta de virtualización por ser un hipervisor de tipo 2, gratuito, multiplataforma y de fácil acceso para estudiantes como nosotras. Su interfaz intuitiva y su compatibilidad con múltiples sistemas operativos permiten crear entornos virtuales replicables sin requerir hardware especializado. Además, al ser software libre, promueve el aprendizaje autónomo y ético en contextos educativos.

2. Marco Teórico

¿Qué es la virtualización?

La virtualización es una tecnología fundamental en arquitectura de computadoras y sistemas operativos. Permite crear entornos aislados que simulan el funcionamiento de una computadora física, utilizando recursos compartidos de forma eficiente y segura.

Es como sembrar múltiples jardines digitales sobre una misma tierra física, porque literalmente, se pueden instalar varias máquinas virtuales sobre una misma máquina física

Según Silberschatz, Galvin y Gagne (2018), la virtualización es “la abstracción de los recursos físicos de una computadora para crear múltiples entornos de ejecución independientes”.

¿Cómo se logra?

Se utiliza un software llamado **hipervisor**, que gestiona las máquinas virtuales (VMs) y su interacción con el hardware físico. Existen dos tipos:

Tipo de Hipervisor	Descripción	Ejemplos
Tipo 1 Bare-metal	Se instala directamente sobre el hardware	VMware ESXi, Microsoft Hyper-V
Tipo 2 Hosted	Se ejecuta sobre un sistema operativo ya instalado	VirtualBox, VMware Workstation

¿Qué tiene una máquina virtual?

Cada VM tiene su propio sistema operativo, memoria, disco y red virtual, **aislados del sistema anfitrión**. Esto permite que funcionen como computadoras independientes dentro de un mismo equipo físico.

¿Qué es un sandbox?

Un **sandbox** es un entorno controlado y aislado donde se pueden ejecutar programas sin correr riesgos, porque no afectan al sistema. En virtualización, cada VM actúa como un sandbox: lo que ocurre dentro no impacta el sistema principal.

La palabra “sand” viene del inglés y significa arena, y “box” de caja, la idea es justamente esa, es como una caja de arena donde se puede experimentar sin riesgo, como en la que juegan los niños.

Ventajas de la virtualización

- **Aislamiento:** cada VM funciona de forma independiente.
- **Seguridad:** si algo falla en una VM, no afecta al sistema anfitrión.
- **Flexibilidad:** se pueden crear, modificar o eliminar entornos fácilmente.
- **Ahorro de recursos:** permite usar un solo equipo físico para múltiples propósitos.
- **Aprendizaje seguro:** ideal para estudiantes que quieren practicar sin dañar su sistema.

Ejemplo aplicado

En este trabajo se utilizará **VirtualBox** (hipervisor tipo 2) para crear una máquina virtual con **Ubuntu Server**, donde se instalará **Python** y se ejecutará un programa como caso práctico. Todo el proceso se realizará en un entorno aislado, replicable y seguro.

3. Caso Práctico

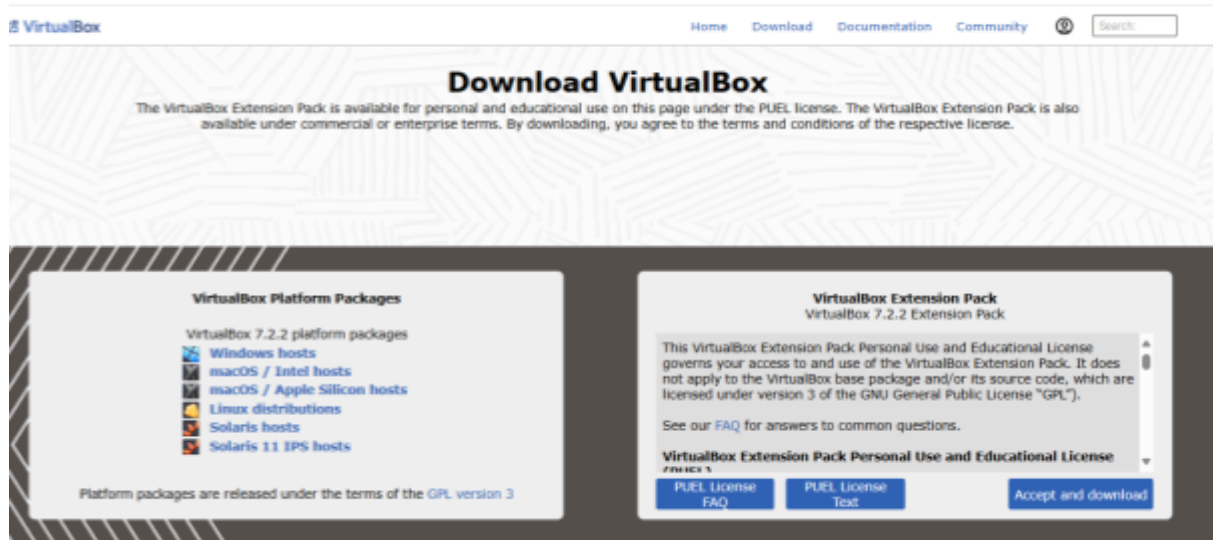
Creación del espacio virtual: instalación del sistema y selección de herramientas

La intención es lograr crear un entorno virtual funcional, que simula un servidor operativo, utilizando herramientas de virtualización y administración de sistemas. Para ello, se emplea **VirtualBox como hipervisor tipo 2**, permitiendo la instalación de **Ubuntu Server 22.04.3 LTS en una máquina virtual**. Este sistema operativo se configura mediante el instalador **curtin**, y se seleccionan herramientas (snaps) que responden a criterios éticos, funcionales y pedagógicos. El entorno resultante será la base para el desarrollo de tareas administrativas, automatización y documentación técnica.

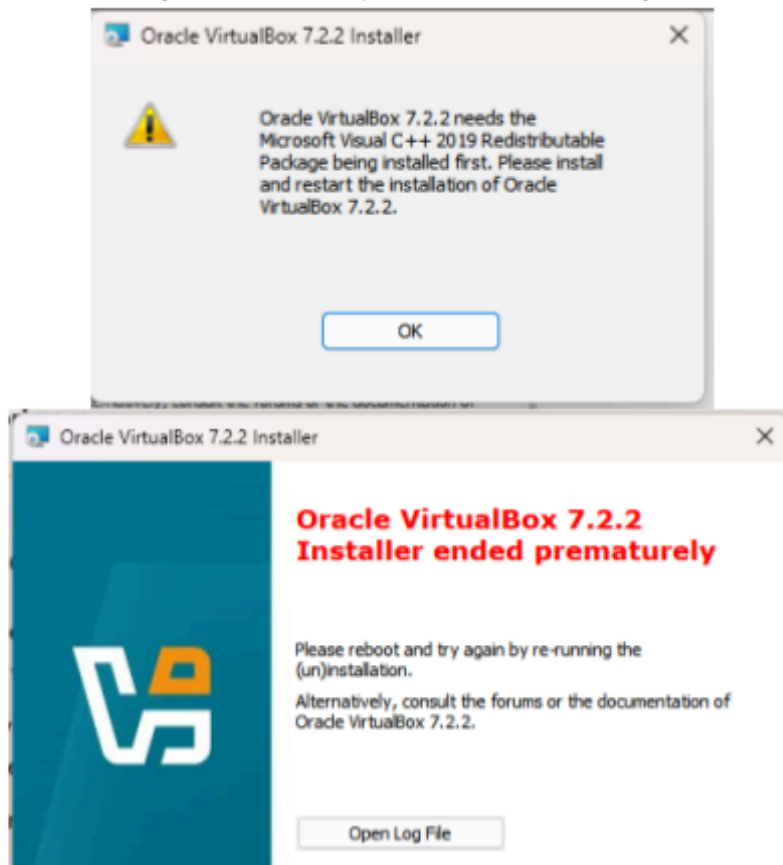
La ventaja de utilizar **Ubuntu Server** es que es libre, transparente, lo que le da autonomía al usuario, permite un entorno eficiente sin interfaz gráfica y es ideal para servidores. Además se registra todo el proceso, paso a paso con las capturas de pantalla. Se investigó, con ayuda de diversas IA, como Copilot y Chat GPT, cuáles eran los snap (herramientas) más recomendadas para respetar la privacidad y no sobrecargar. Preferimos utilizar VirtualBox como hipervisor tipo 2, porque permite aislar el entorno sin afectar al sistema anfitrión haciendo que sea un proceso de virtualización segura.

Se incluyen capturas de la instalación, selección de snaps y cierre del sistema, documentadas con frases de anclaje que acompañen la lectura del proceso realizado. También se documentan los ajustes y correcciones lo consideramos importante como parte de todo el proceso, respeta además la realidad de cada paso que fuimos dando, para poder instalar el sistema se requería la instalación de C++

Captura 1 – Elección de hipervisor, inicio del entorno virtual



Captura 2 – Interrupción del entorno, instalación fallida del hipervisor



Captura 3 - Resolviendo problemas para instalación de VirtualBox

Version: Visual Studio 2022

Filter by title

Latest Supported Visual C++ Redistributable Downloads

How to audit Visual C++ Runtime version usage

> Deployment examples

Redistribute web client applications

ClickOnce deployment for Visual C++ applications

Run a C++ -clr application with previous runtime version

C++ Attributes for COM Attribute programming

> Attributes by group

> Attributes by usage

Download PDF

Latest supported redistributable version

Expand table

Architecture	Link	Notes
ARM64	https://aka.ms/vs/17/release/vc_redist.arm64.exe	Permalink for latest supported ARM64 version
x86	https://aka.ms/vs/17/release/vc_redist.x86.exe	Permalink for latest supported x86 version
x64	https://aka.ms/vs/17/release/vc_redist.x64.exe	Permalink for latest supported x64 version. The X64 Redistributable package

Instalación de Microsoft Visual C++ 2015-2022 Redistributable (x64) - 14.44.35211

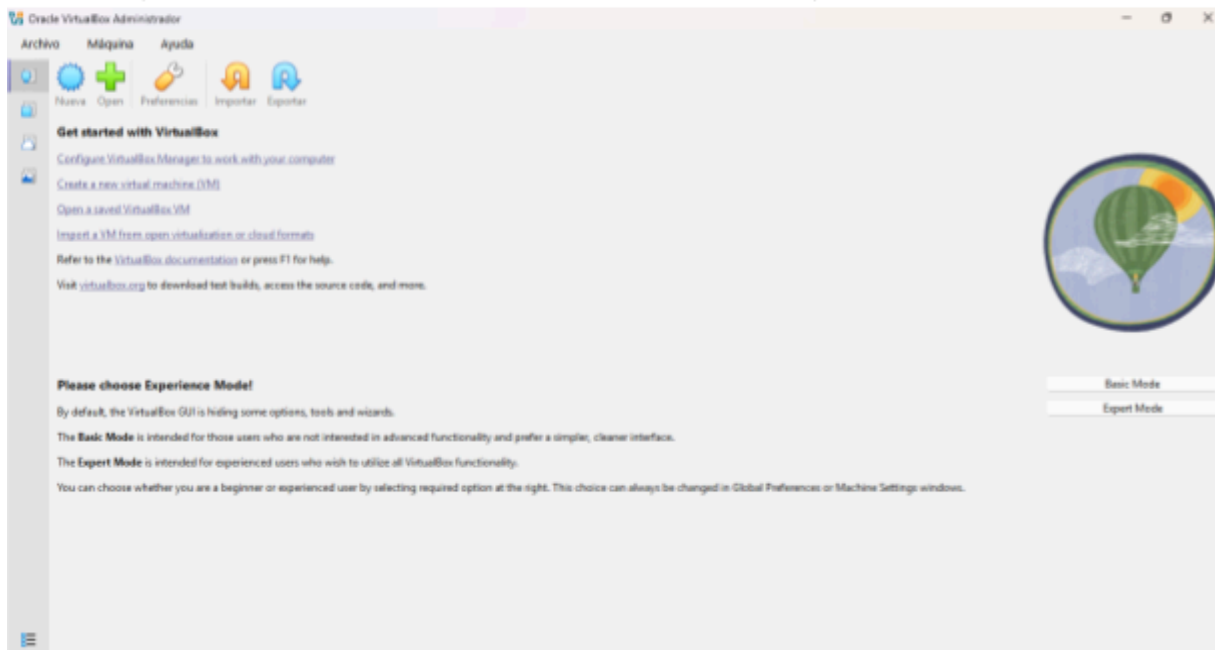
Microsoft Visual C++ 2015-2022 Redistributable (x64) - 14.44.35211

Progreso de la instalación

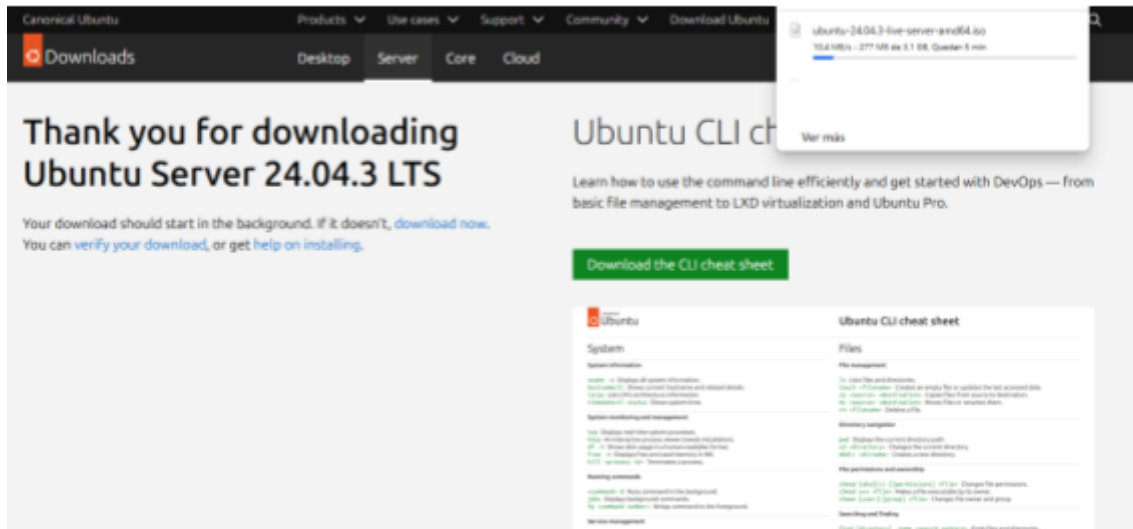
Procesando: Microsoft Visual C++ 2022 X64 Additional Runtime - 14.44.35211

Cancelar

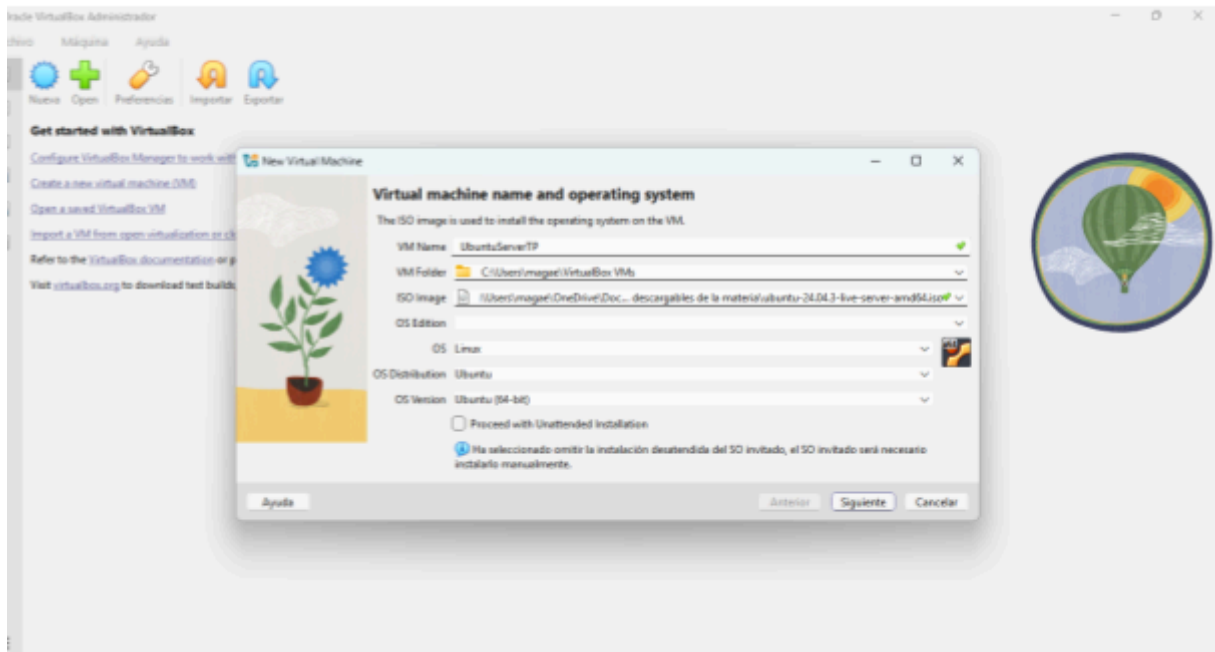
Captura 4 – Inicio del entorno técnico, VirtualBox instalado y listo



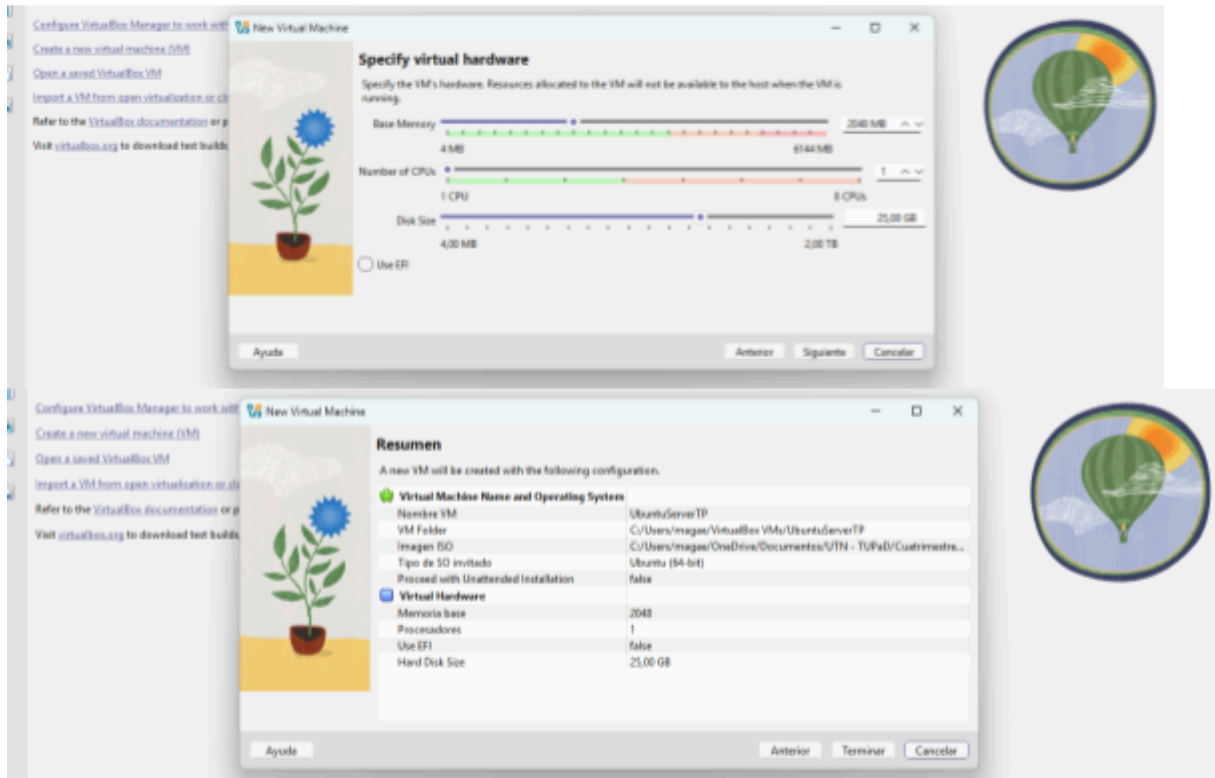
Captura 5 – Descarga del sistema base, Ubuntu Server



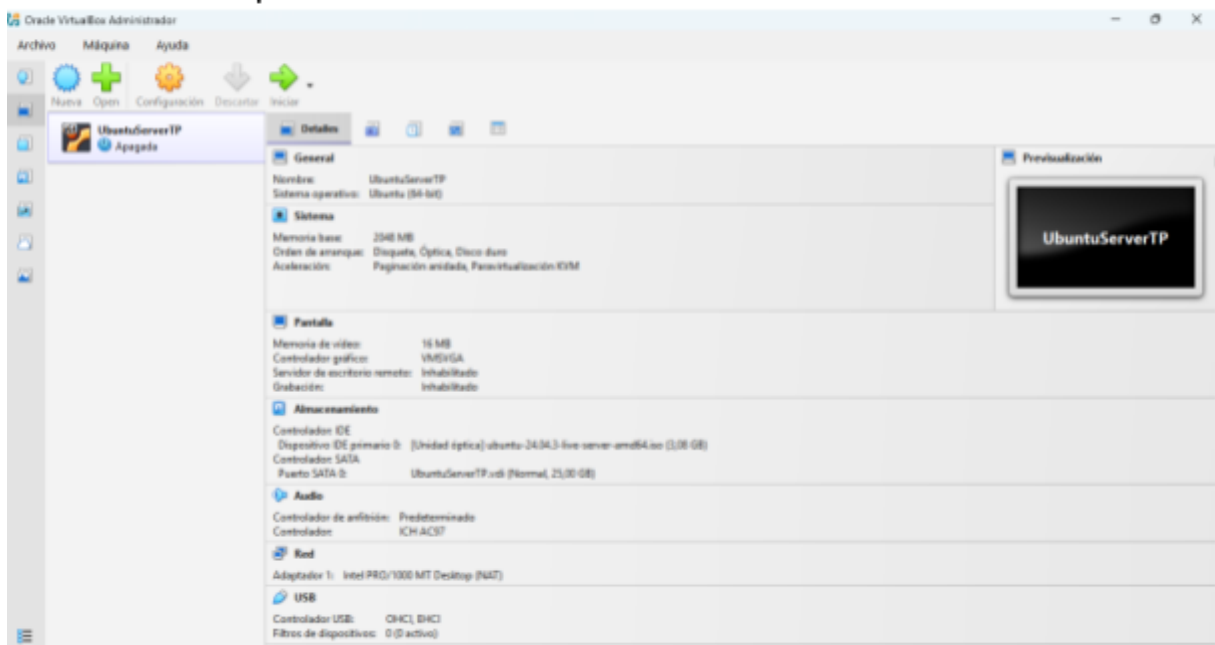
Captura 6 – Configuración inicial de la máquina virtual



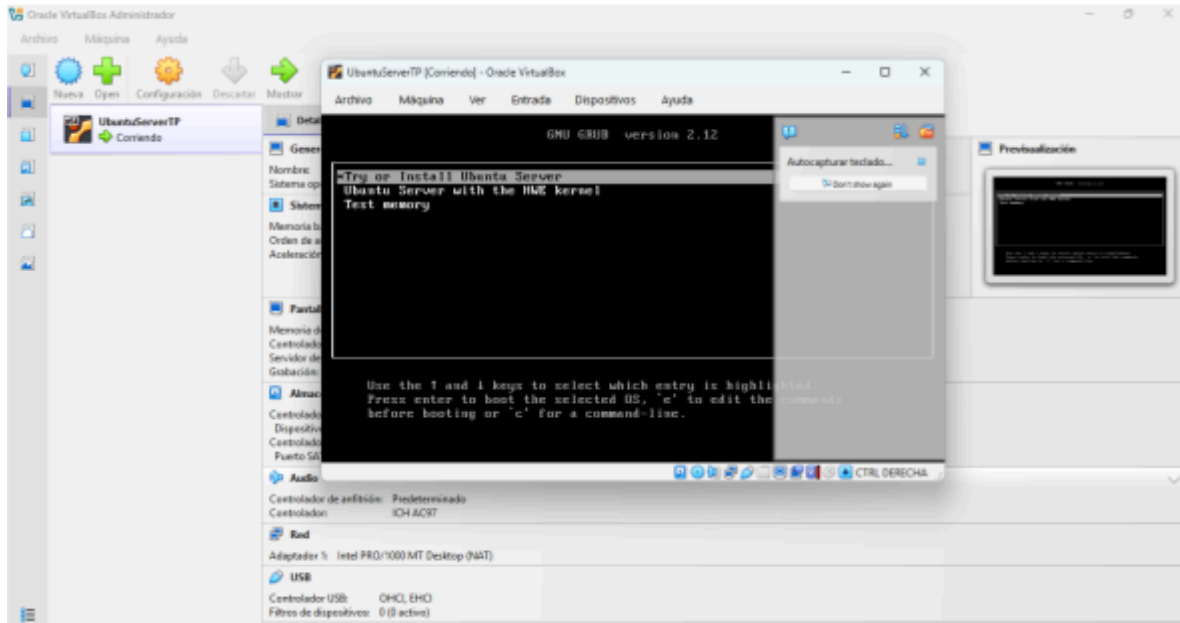
Captura 7 – Confirmación de recursos técnicos



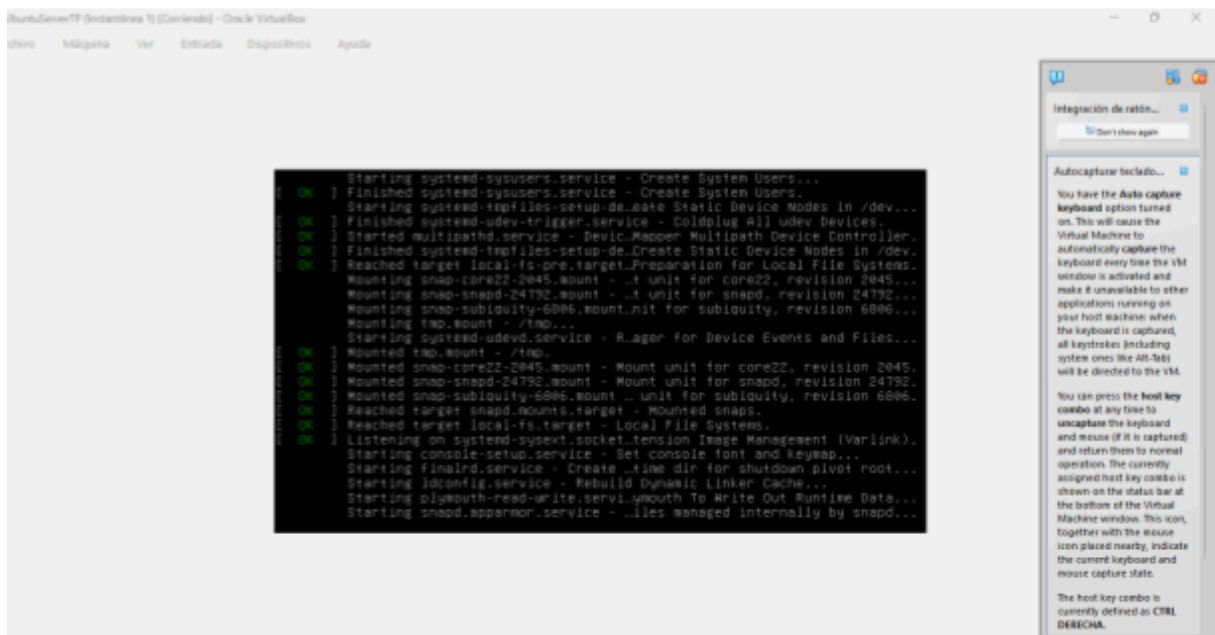
Captura 8 – Revisión final antes del inicio



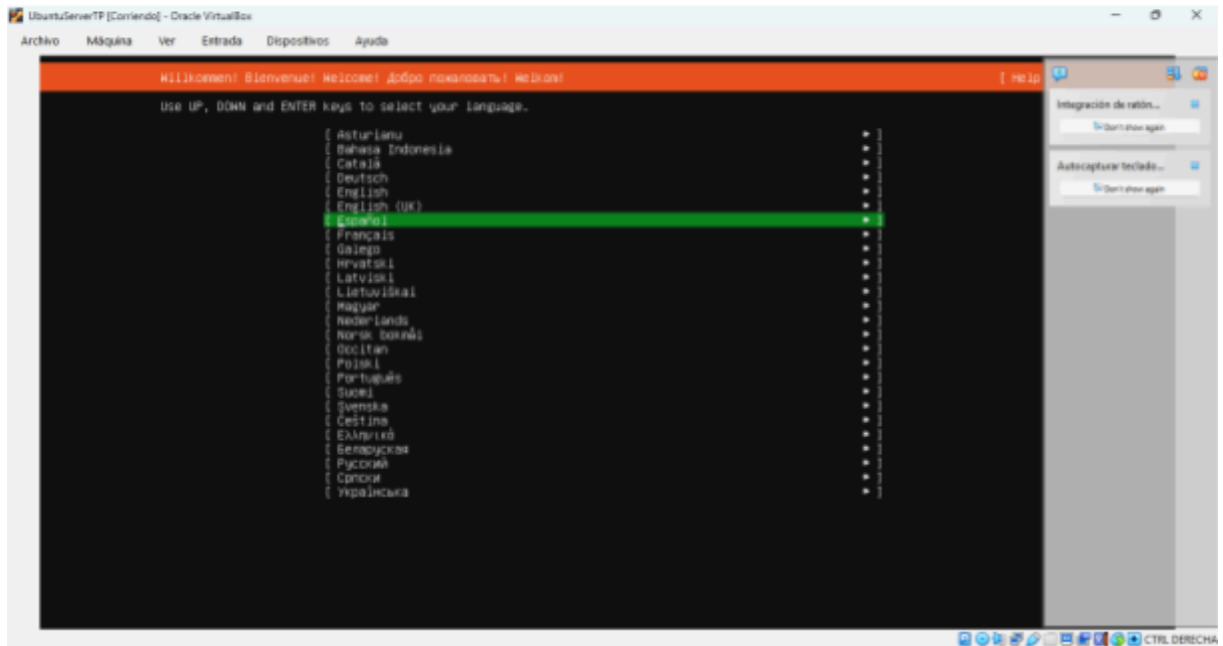
Captura 9 – GRUB: Inicio del instalador



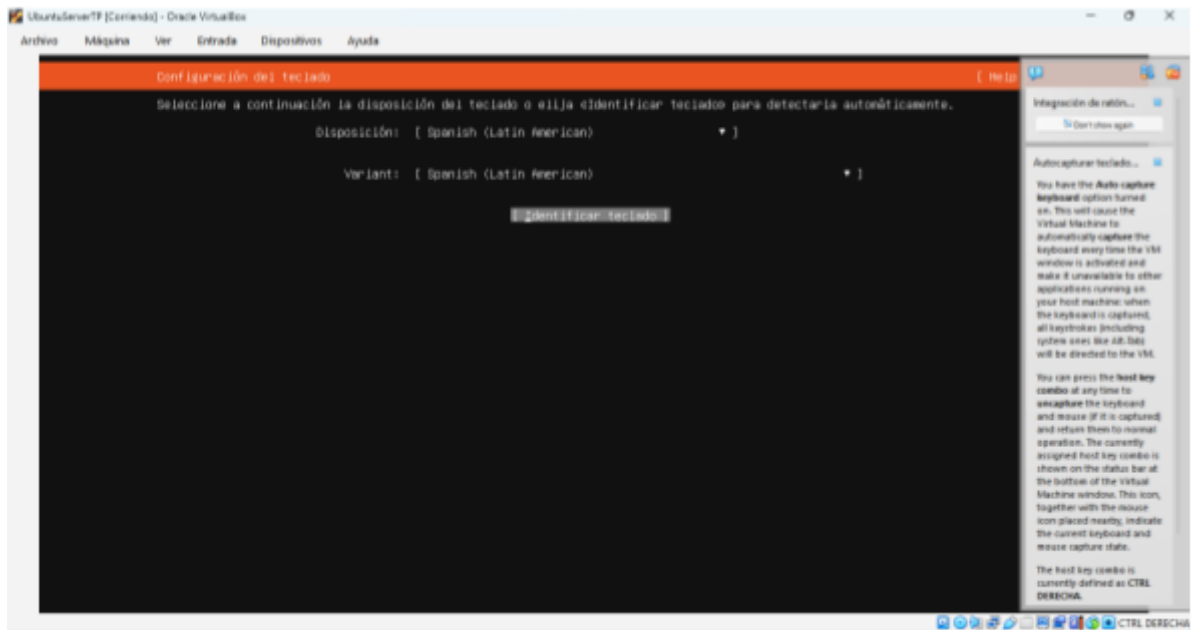
Captura 9.1 - Instalando



Captura 10 - Configurando el sistema, eligiendo el idioma

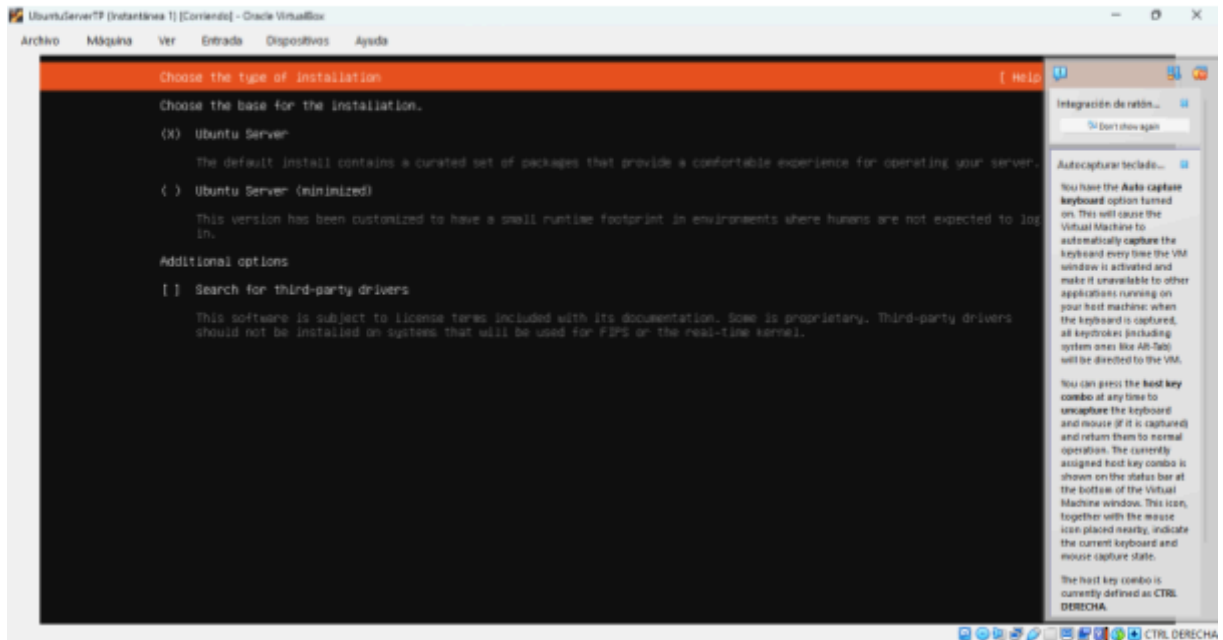


Captura 11 - Configuración del Teclado



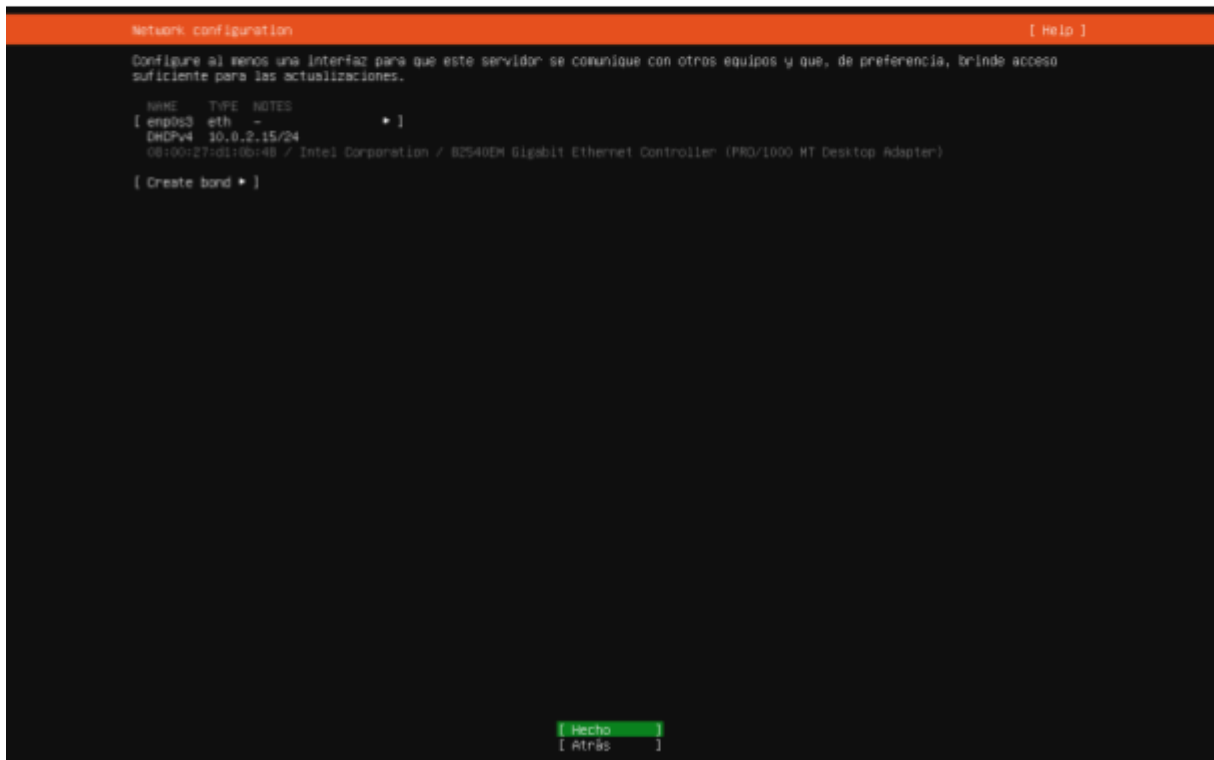
Captura 12 – Elección del tipo de instalación

UbuntuServerTP – Elección de instalación: entorno completo o minimizado

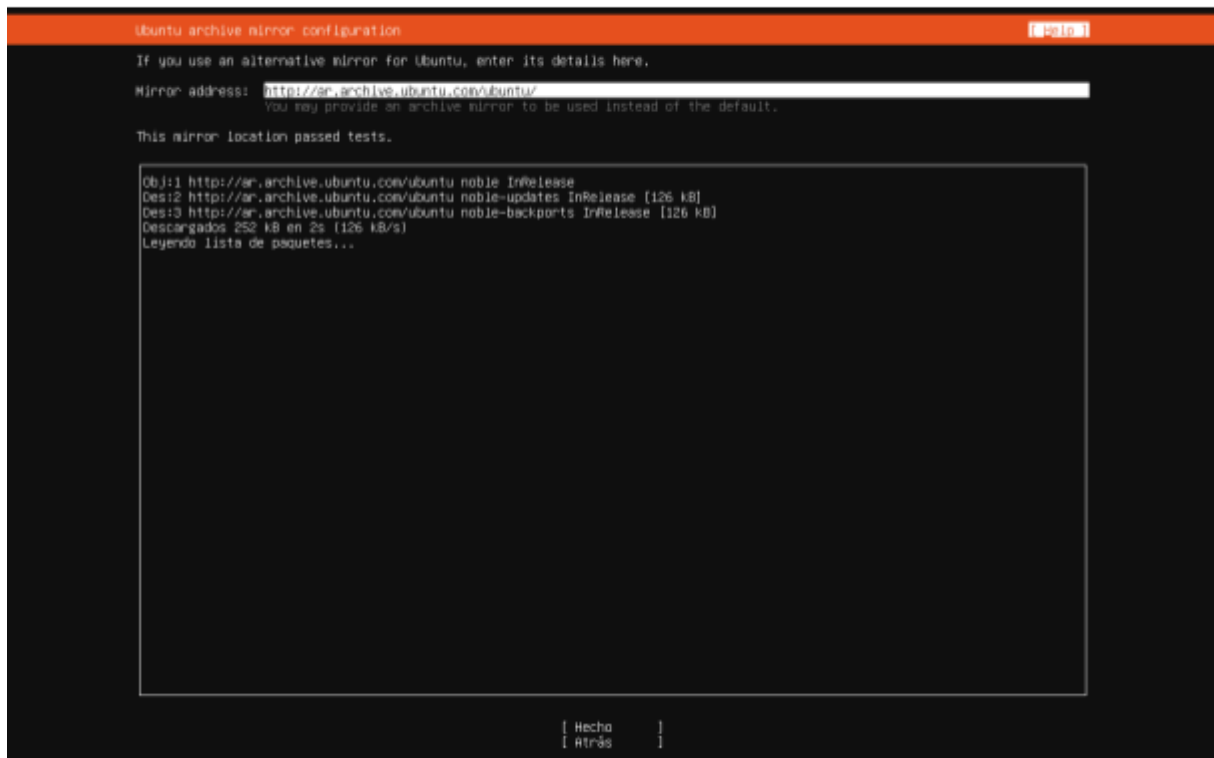


Captura 13 – Conexión del sistema al entorno

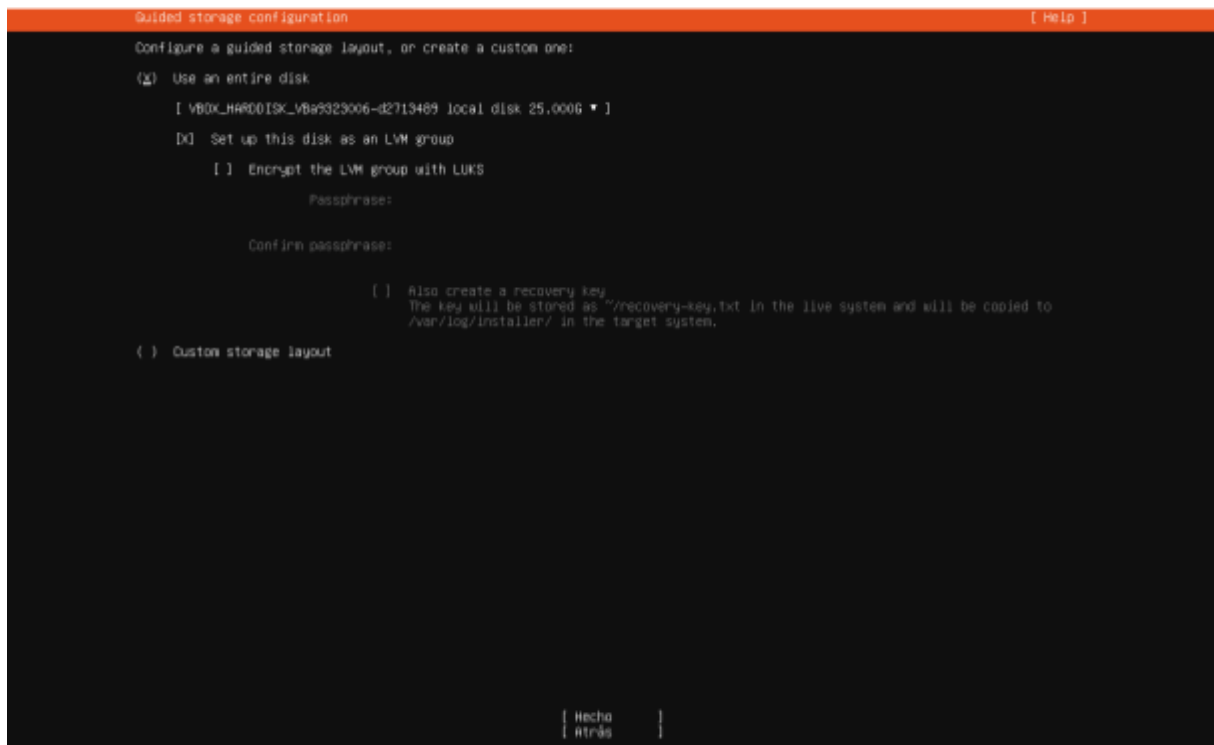
Configuración de red: raíz digital del sistema



Captura 14 - Configuración de Espejo de Paquetes



Captura 15 - Particionando el disco



Captura 16 - Confirmación de Particionado

Storage configuration [Help]

RESUMEN DEL SISTEMA DE ARCHIVOS

PUNTO DE MONTAJE	TAMANO	TIPO	TIPO DE DISPOSITIVO
/	11.496G	new ext4	new LVM logical volume ▶
/boot	2.000G	new ext4	new partition of disco local ▶

DISPOSITIVOS DISPONIBLES

DISPOSITIVO	TIPO	TAMANO
ubuntu-vg (new)	LVM volume group	22.996G ▶
espacio disponible		11.500G ▶

[Create software RAID (md) ▶]
[Crear grupo de volúmenes (LVM) ▶]

DISPOSITIVOS UTILIZADOS

DISPOSITIVO	TIPO	TAMANO
ubuntu-vg (new)	LVM volume group	22.996G ▶
ubuntu-lv	new, to be formatted as ext4, mounted at /	11.496G ▶
VBDK_HARDDISK_VBa9323006-d2713489	disco local	25.000G ▶
partition 1	new, BIOS grub spacer	1.000M ▶
partition 2	new, to be formatted as ext4, mounted at /boot	2.000G ▶
partition 3	new, PV of LVM volume group ubuntu-vg	22.997G ▶

[Hecho]
[Restablecer]
[Atrás]

Captura 17 - Confirmación para formatear

Storage configuration [Help]

RESUMEN DEL SISTEMA DE ARCHIVOS

PUNTO DE MONTAJE	TAMANO	TIPO	TIPO DE DISPOSITIVO
/	11.496G	new ext4	new LVM logical volume ▶
/boot	2.000G	new ext4	new partition of disco local ▶

DISPOSITIVOS DISPONIBLES

DISPOSITIVO	TIPO	TAMANO
ubuntu-vg (new)	LVM volume group	22.996G ▶
espacio disponible		11.500G ▶

[Create software RAID (md) ▶]
[Crear grupo de volúmenes (LVM) ▶]

DISPOSITIVOS UTILIZADOS

DISPOSITIVO	TIPO	TAMANO
ubuntu-vg (new)		
ubuntu-lv	n	
VBDK_HARDDISK_VBa9323		
partition 1	n	
partition 2	n	
partition 3	n	

Confirmar acción destructiva

Selecting Continue below will begin the installation process and result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the installation has started.

Are you sure you want to continue?

[No]
[Continuar]

[Hecho]
[Restablecer]
[Atrás]

Captura 18 - Identificación del Sistema

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 server's name:
The name it uses when it talks to other computers.

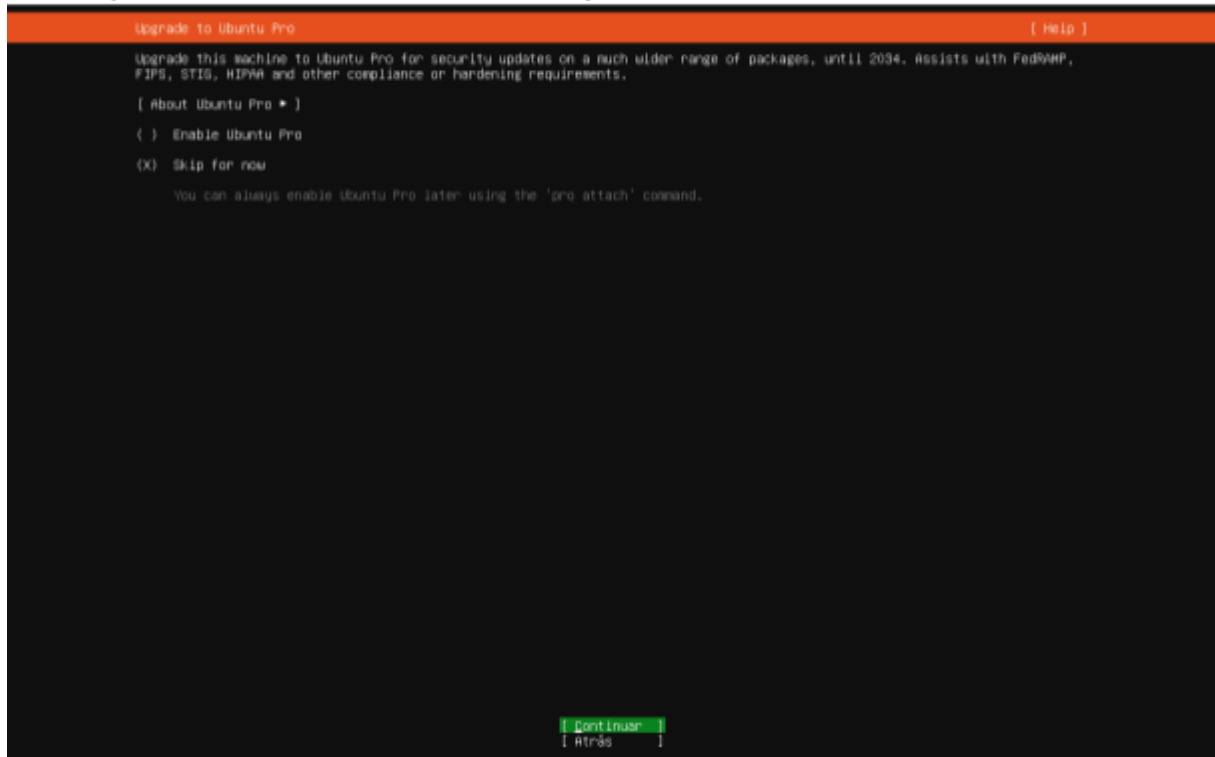
Elija un nombre de usuario:

Elija una contraseña:

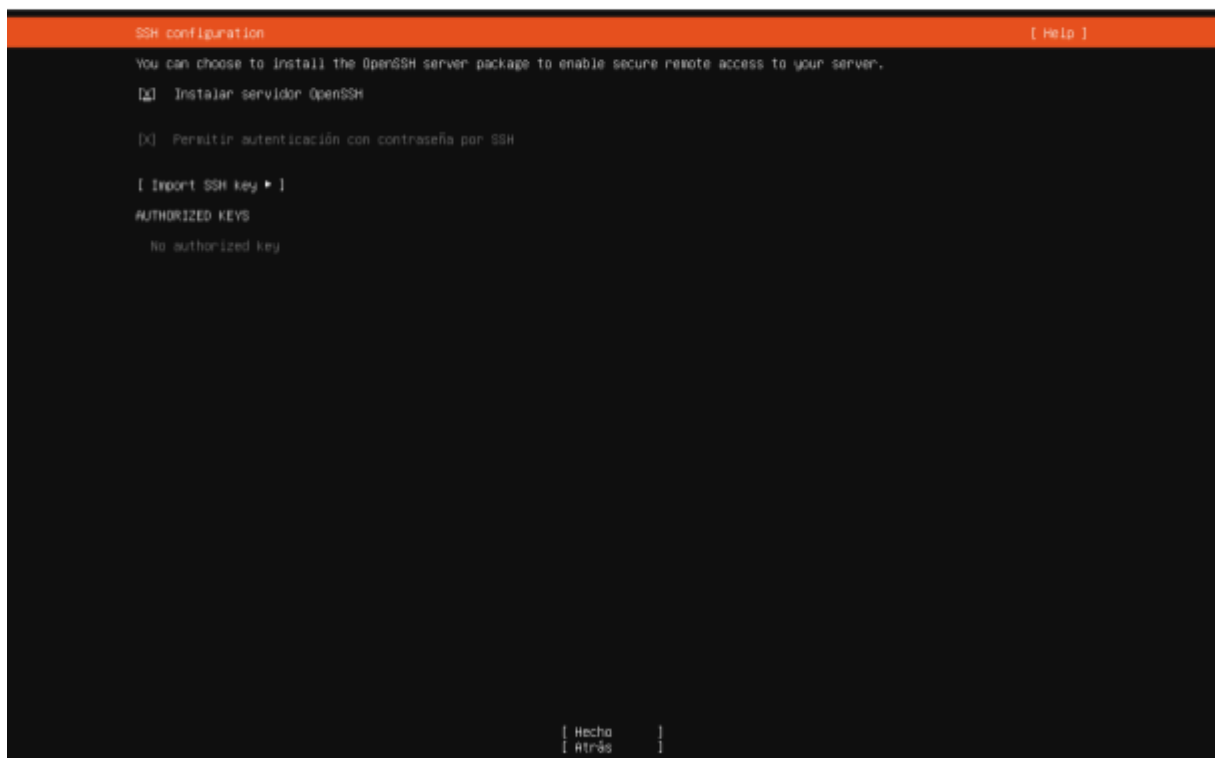
Confirme la contraseña:

[Hecho]

Captura 19 – Ubuntu Pro: decisión sobre soporte extendido



Captura 20 – Configuración de SSH: acceso remoto



Captura 21 – Snaps destacados

```

Featured server snaps                                     ( Help )

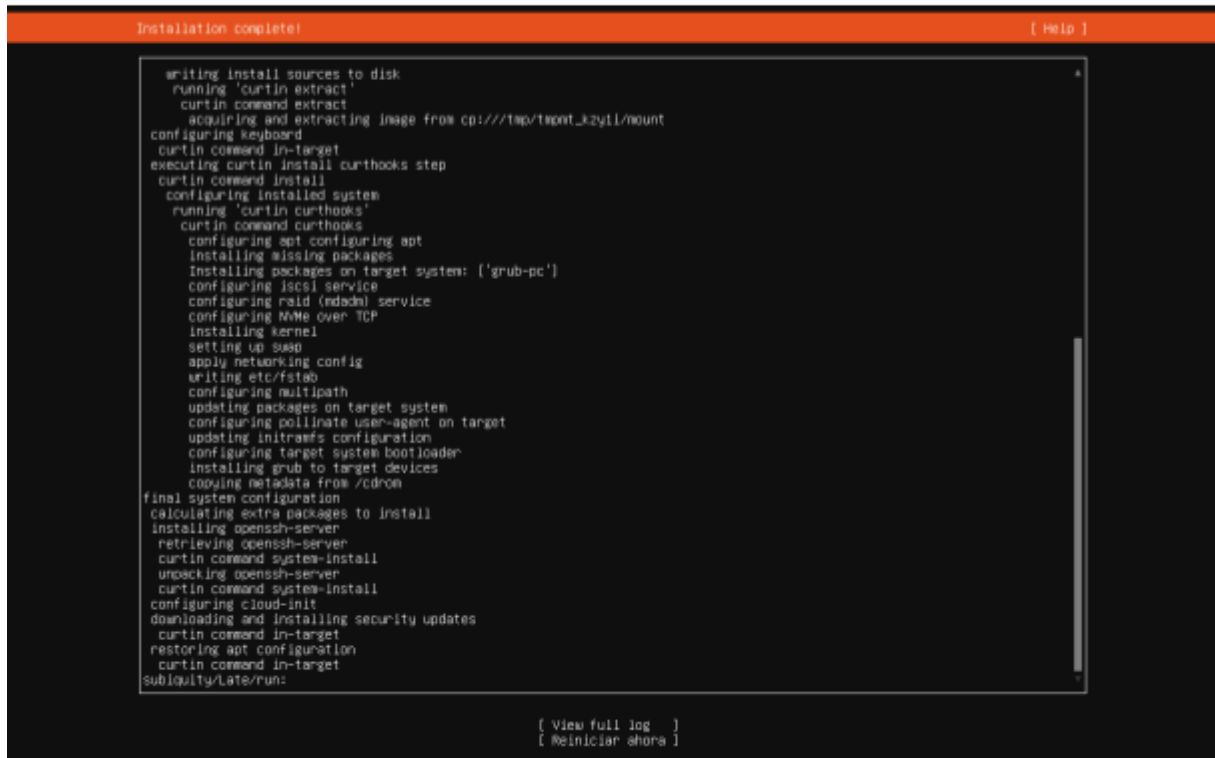
These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see more details of the package,
publisher and versions available.

[ ] 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-livepatch canonical      Canonical Livepatch Client
[*] rocket.chat-server rocket.chat✓   rocket.chat server
[ ] mosquitto         mosquitto✓    Eclipse Mosquitto MQTT broker
[ ] etcd              canonical✓    Resilient key-value store by CoreOS
[ ] powershell        canonical✓    PowerShell for every system!
[ ] sdnzbd            safihr       sdnzbd
[ ] wormhole          snapcrafters 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      amd✓         PostgreSQL is a powerful, open source object-relational database system.
[ ] keepalived         keepalived-project✓ High availability VRRP/80 and load-balancing for Linux
[ ] prometheus        canonical✓    The Prometheus monitoring system and time series database
[ ] lxd               canonical✓    LXD - container and VM manager

[ Hacho      ]
[ Atrás      ]

```

Captura 22 – Instalación Completa



```
Installation complete! [ Help ]

writing install sources to disk
  running 'curtin extract'
  curtin command extract
    acquiring and extracting image from cp:///tmp/tpmnt_kzytl/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: ['grub-pc']
      configuring iscsi service
      configuring raid (mdadm) service
      configuring NvMe over TCP
      installing kernel
      setting up suseo
      apply networking config
      writing etc/fstab
      configuring multipath
      updating packages on target system
      configuring pollinate user-agent on target
      updating initramfs configuration
      configuring target system bootloader
      installing grub to target devices
      copying metadata from /cdrom
final system configuration
  calculating extra packages to install
  installing openssh-server
  retrieving openssh-server
  curtin command system-install
  unpacking openssh-server
  curtin command system-install
  configuring cloud-init
  downloading and installing security updates
  curtin command in-target
  restoring apt configuration
  curtin command in-target
subiquity/late/run:

[ View full log ]
[ Reiniciar ahora ]
```

La instalación se completó correctamente mediante el instalador **curtin**, y el sistema está listo para iniciar sesión con el **usuario luz_magaly**. Se seleccionaron herramientas como nginx, postgresql y glances, validadas por su utilidad en el TP integrador.

Inicio de Sesión en Ubuntu Server

Breve descripción técnica:

Ya iniciado el sistema desde VirtualBox, se accede con el usuario creado: luz_magaly y se ejecutan algunos comandos básicos para validar que el entorno funcione correctamente. Se verifica el estado del servicio snapd, el uso de recursos del sistema, y la conectividad de red. Esta etapa confirma que el servidor está operativo y listo para recibir tareas técnicas.

Comandos ejecutados:

Comando	¿Para qué?
<code>sudo apt update && sudo apt list --upgradable</code>	Verifica actualizaciones disponibles
<code>sudo systemctl is-enabled snapd</code>	Comprueba si el servicio está habilitado
<code>sudo systemctl is-active snapd</code>	Verifica si el servicio está activo
<code>sudo systemctl status snapd</code>	Muestra el estado detallado del servicio
<code>df -h</code>	Muestra el uso del disco
<code>ip a</code>	Verifica la interfaz de red

Los **snaps** son los paquetes de software que se instalaron, para poder verificarlos se utilizan los **snapd** que son servicios del sistema que instalan, actualizan y ejecutan los snaps

Arranque del Sistema e instalación de Herramientas

Captura 23.1 - Arranque y Chequeo del Sistema: fsck y limpieza de inodos

```
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
Begin: Running /scripts/local-premount ... [ 3.831850] Btrfs loaded, zoned=yes
s, fsverity=yes
Scanning for Btrfs filesystems
done.
Begin: Will now check root file system ... fsck from util-linux 2.39.3
[/usr/sbin/fsck.ext4 (1) -- /dev/mapper/ubuntu--vg-ubuntu--lv] fsck.ext4 -a -C0
/dev/mapper/ubuntu--vg-ubuntu--lv
/dev/mapper/ubuntu--vg-ubuntu--lv: recovering journal
/dev/mapper/ubuntu--vg-ubuntu--lv: Clearing orphaned inode 393284 (uid=0, gid=0,
mode=0100600, size=0)
/dev/mapper/ubuntu--vg-ubuntu--lv: Clearing orphaned inode 393283 (uid=0, gid=0,
mode=0100600, size=0)
/dev/mapper/ubuntu--vg-ubuntu--lv: Clearing orphaned inode 393282 (uid=0, gid=0,
mode=0100600, size=0)
/dev/mapper/ubuntu--vg-ubuntu--lv: Clearing orphaned inode 393267 (uid=0, gid=0,
mode=0100600, size=0)
/dev/mapper/ubuntu--vg-ubuntu--lv: clean, 98001/753664 files, 1419499/3013632 bl
ocks
done.
[ 4.157068] EXT4-fs (dm-0): mounted filesystem a57247c5-dc5e-40f3-8e94-e4d34d
9393f3 ro with ordered data mode. Quota mode: none.
done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ...
```

Captura 23.2 - Arranque y Chequeo del Sistema: Servicios iniciándose

```
OK ] Starting snapd.socket - Socket activation for snappy daemon...
OK ] Listening on ssh.socket - OpenSSH Secure Shell server socket.
OK ] Listening on uidd.socket - UID daemon activation socket.
OK ] Listening on lxd-installer.socket - Helper to install lxd snap on demand.
OK ] Listening on snapd.socket - Socket activation for snappy daemon.
OK ] Reached target sockets.target - Socket Units.
OK ] Reached target basic.target - Basic System.
Starting dbus.service - D-Bus System Message Bus...
OK ] Started dbus.service - Save initial kernel messages after boot.
Starting e2scrub_reap.service - Remove Stale Online ext4 Metadata Check Snapshots...
Starting grub-common.service - Record successful boot for GRUB...
Starting polkit.service - Authorization Manager...
Starting rsyslog.service - System Logging Service...
OK ] Started snap.nextcloud.apache.service - Service for snap application nextcloud.apache.
OK ] Started snap.nextcloud.mysql.service - Service for snap application nextcloud.mysql.
OK ] Started snap.nextcloud.nextcloud-cron.service - Service for snap application nextcloud.nextcloud-cron.
OK ] Started snap.nextcloud.nextcloud-fixer.service - Service for snap application nextcloud.nextcloud-fixer.
OK ] Started snap.nextcloud.php-fpm.service - Service for snap application nextcloud.php-fpm.
OK ] Started snap.nextcloud.redis-server.service - Service for snap application nextcloud.redis-server.
OK ] Started snap.nextcloud.renew-certs.service - Service for snap application nextcloud.renew-certs.
OK ] Reached target getty-pre.target - Preparation for Login.
Starting snap.seeded.service - Wait until snapd is fully seeded...
Starting snapd.service - Snap Daemon...
Starting sysstat.service - Resets System Activity Logs...
Starting systemd-logind.service - User Login Management...
Starting udisks2.service - Disk Manager...
OK ] Started dbus.service - D-Bus System Message Bus.
OK ] Finished e2scrub_reap.service - Remove Stale Online ext4 Metadata Check Snapshots.
OK ] Finished sysstat.service - Resets System Activity Logs.
OK ] Finished systemd-networkd-wait-online.service - Wait for Network to be Configured.
OK ] Started systemd-logind.service - User Login Management.
OK ] Reached target network-online.target - Network is Online.
OK ] Started update-notifier-download.timer - Download data for packages that failed at package install time.
OK ] Started update-notifier-notd.timer - Check to see whether there is a new version of Ubuntu available.
OK ] Reached target timers.target - Timer Units.
OK ] Reached target remote-fs-pre.target - Preparation for Remote File Systems.
OK ] Reached target remote-fs.target - Remote File Systems.
Starting apport.service - automatic crash report generation...
OK ] Finished blk-availability.service - Availability of block devices.
OK ] Started cron.service - Regular background program processing daemon.
Starting systemd-user-sessions.service - Permit User Sessions...
OK ] Started unattended-upgrades.service - Unattended Upgrades Shutdown.
OK ] Finished grub-common.service - Record successful boot for GRUB.
OK ] Started rsyslog.service - System Logging Service.
Starting grub-initrd-fallback.service - GRUB failed boot detection...
OK ] Finished systemd-user-sessions.service - Permit User Sessions.
Starting plymouth-quit-wait.service - Hold until boot process finishes up...
Starting plymouth-quit.service - Terminate Plymouth Boot Screen...
OK ] Started polkit.service - Authorization Manager.
```

Captura 24 - Login

```
ubuntu 24.04.3 LTS Integradorayso tty1
Integradorayso login: _
```

Captura 25 - Inicio de Sistema

```
ubuntu 24.04.3 LTS Integradorayso tty1
Integradorayso login: luz_nagaly
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-85-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/so

System information as of mié 15 oct 2025 16:39:31 UTC

System load:          1.0
Usage of /:            43.1% of 11.21GB
Memory usage:         10%
Swap usage:           0%
Processes:            100
Users logged in:       0
IPv4 address for enp0s3: 10.0.2.15
IPv6 address for enp0s3: fd17:625c:f037:2:a00:27ff:fed1:b40

El mantenimiento de seguridad expandido para Applications está desactivado.
Se pueden aplicar 26 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

luz_nagaly@Integradorayso:~$ sudo _
```

Durante el arranque del sistema, Ubuntu Server ejecuta una verificación del sistema de archivos (fsck) y limpia estructuras huérfanas. Después inicia los servicios esenciales y prepara el entorno para el ingreso del usuario. Las capturas muestran que el sistema se ordena, respira y está listo para recibir tareas técnicas.

Luego se instalan herramientas esenciales usando el gestor snap. En la captura se ve el sistema activo, con comandos para instalar glances y nextcloud, y mensajes que confirman que el entorno está funcionando correctamente.

Captura 26 - Instalando Herramientas

```

ubuntu 24.04.3 LTS integradorayso tty1
integradorayso login: luz_magaly
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-85-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of mié 15 oct 2025 17:21:23 UTC

System load:          0.20
Usage of /:            45.9% of 11.21GB
Memory usage:         19%
Swap usage:           0%
Processes:            114
Users logged in:      0
IPv4 address for enp0s3: 10.0.2.15
IPv6 address for enp0s3: fd17:625c:f037:2:a00:27ff:fed1:b48

El mantenimiento de seguridad expandido para Applications está desactivado.
Se pueden aplicar 26 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute 'sudo pro status'

To run a command as administrator (user "root"), use "sudo <command>".
See 'man sudo_root' for details.

luz_magaly@integradorayso:~$ sudo snap install glances
[sudo] password for luz_magaly:
glances 4.3.3+build01 from nicolargo installed
luz_magaly@integradorayso:~$ sudo snap install nextcloud
snap "nextcloud" is already installed, see 'snap help refresh'
luz_magaly@integradorayso:~$
luz_magaly@integradorayso:~$

```

Captura 27 - Snap List: Listado de herramientas

```

ubuntu 24.04.3 LTS integradorayso tty1
integradorayso login: luz_magaly
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-85-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of mié 15 oct 2025 17:21:23 UTC

System load:          0.20
Usage of /:            45.9% of 11.21GB
Memory usage:         19%
Swap usage:           0%
Processes:            114
Users logged in:      0
IPv4 address for enp0s3: 10.0.2.15
IPv6 address for enp0s3: fd17:625c:f037:2:a00:27ff:fed1:b48

El mantenimiento de seguridad expandido para Applications está desactivado.
Se pueden aplicar 26 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute 'sudo pro status'

luz_magaly@integradorayso:~$ snap list
Name      Version    Rev    Tracking    Publisher    Notes
core18    20250710   2352   latest/stable    canonical+   base
core24    20250829   1196   latest/stable    canonical+   base
glances   4.3.3+build01 2331   latest/stable    nicolargo    -
nextcloud 31.0.9snap1 50464   latest/stable    nextcloud+   -
snapd     2.71       25282   latest/stable    canonical+   snapd
luz_magaly@integradorayso:~$

```

Se ejecuta el comando `snap list` para verificar los paquetes Snap instalados. El sistema muestra los componentes base (core, core18) y el servicio `snappy`, confirmando que el entorno está preparado para instalar herramientas como `glances` y `nextcloud`. Esta validación técnica respalda la autonomía funcional del servidor

Aquí se debe presentar un problema o situación concreta que haya sido desarrollada o simulada para aplicar el contenido del trabajo.

Instalación de Python

Captura 27 - Instalación de Python

```
ubuntu 24.04.3 LTS integradorayso tty1
integradorayso login: luz_magaly
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-85-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of 2025-10-15 17:21:23 UTC

System load:          0.28
Usage of /:            45.9% of 11.2GB
Memory usage:         19%
Swap usage:            0%
Processes:             114
Users logged in:       0
IPV4 address for enp0s3: 10.0.2.15
IPV6 address for enp0s3: fd17:625c:f037:2::a00:27ff:fed1:b40

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 26 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute 'sudo pro status'

luz_magaly@integradorayso:~$ sudo apt install python3
[sudo] password for luz_magaly:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
python3 ya está en su versión más reciente (3.12.3-0ubuntu2).
Fijado python3 como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 26 no actualizados.
luz_magaly@integradorayso:~$
```


Captura 28 - Creación del Script con Nano

```
#!/bin/bash #
# Actividad, ingresar los nombres de 3 alumnos, y para cada uno una tupla de 3 notas.
# Luego, mostrar el promedio de cada alumno.

print("Hola!\n\n")

# Validación de que un valor sea un número

lista_completa_notas=()
promedios=()

def probar_numero(n):
    try:
        n=float(n)
        return True
    except ValueError:
        return False

# Calcular promedio de cada estudiante
def calcular_promedio(lista):
    suma=0
    for nota in lista:
        suma+=nota
    return suma / len(lista)

# Solicitar los tres nombres y notas
for i in range(3):
    # Solicitar el nombre del alumno al usuario y validar que no esté vacío
    nombre=input("¿Cuál es el nombre del estudiante que querés agregar?: ").strip().title()
    while nombre=="":
        print("\nEl nombre no puede estar vacío, vuelve a intentarlo...")
        nombre=input("¿Cuál es el nombre del estudiante que querés agregar?: ").strip().title()

    # Solicitar las notas de cada alumno y armar la tupla
    lista_notas=()
    for j in range(1,4):
        nota=1
        while nota<1 or nota>10:
            nota_texto=input(f"Nota {j} de {nombre}: ").strip()
            while not probar_numero(nota_texto):
                print("\nDebes ingresar un valor numérico, intenta nuevamente...")
            nota_texto=input(f"Nota {j} de {nombre}: ").strip()
            nota=float(nota_texto)
            if nota<1 or nota>10:
                print("La nota debe ser un valor entre 1 y 10, intenta nuevamente...")
            lista_notas.append(nota)
        tupla_notas=tuple(lista_notas)
```

Captura 29 - Ejecución del Script

```
Nota 1 de Ana María: -5
La nota debe ser un valor entre 1 y 10, intenta nuevamente...
Nota 1 de Ana María: 4352
La nota debe ser un valor entre 1 y 10, intenta nuevamente...
Nota 1 de Ana María: 7
Nota 2 de Ana María:
Debes ingresar un valor numérico, intenta nuevamente...
Nota 2 de Ana María: 8
Nota 3 de Ana María: 9

Las notas de Ana María son (7.0, 8.0, 9.0)

El promedio de las notas de Ana María es 8.00

¿Cuál es el nombre del estudiante que querés agregar?:

El nombre no puede estar vacío, vuelve a intentarlo...
¿Cuál es el nombre del estudiante que querés agregar?: pedro jose garcia
Nota 1 de Pedro Jose Garcia: 5.6
Nota 2 de Pedro Jose Garcia: 7
Nota 3 de Pedro Jose Garcia: 9

Las notas de Pedro Jose Garcia son (5.6, 7.0, 9.0)

El promedio de las notas de Pedro Jose Garcia es 7.20

¿Cuál es el nombre del estudiante que querés agregar?: monica azul PEREZ
Nota 1 de Monica Azul Perez: 7
Nota 2 de Monica Azul Perez: 8
Nota 3 de Monica Azul Perez: 4

Las notas de Monica Azul Perez son (7.0, 8.0, 4.0)

El promedio de las notas de Monica Azul Perez es 6.33

Las notas registradas son:
Ana María: (7.0, 8.0, 9.0)
Pedro Jose Garcia: (5.6, 7.0, 9.0)
Monica Azul Perez: (7.0, 8.0, 4.0)

Los promedios registrados son:
Ana María: 8.0
Pedro Jose Garcia: 7.2
Monica Azul Perez: 6.33

¡Muchas gracias y hasta luego!

luz.nagalv@integradora.ego:/mnt$
```

4. Metodología Utilizada

Investigación previa (fuentes utilizadas).

Se consultaron fuentes como el material de las clases (Arquitectura y Sistemas Operativos, TUPaF UTN), la documentación oficial de Ubuntu, foros técnicos, y herramientas de inteligencia artificial como Copilot y ChatGPT. Estas fuentes ayudaron a seleccionar herramientas éticas, funcionales y livianas, como glances, nextcloud, nginx y postgresql, priorizando la autonomía del entorno y la privacidad del usuario.

Etapas de diseño y prueba del código.

- Instalación de VirtualBox como hipervisor tipo 2
- Creación de la máquina virtual con Ubuntu Server 20.04.3 LTS
- Instalación del sistema mediante el instalador curtin
- Validación del entorno con comandos (snap list, systemctl, df -h, ip a)
- Instalación de herramientas Snap
- Captura de pantalla de cada etapa
- Documentación simbólica y técnica del proceso

Herramientas y recursos utilizados (IDE, librerías, control de versiones, etc.).

- VirtualBox
- Ubuntu Server
- Snapd
- Python
- Glances
- Nextcloud
- Capturas de pantalla
- Documentación colaborativa en Word
- IA como Copilot y ChatGPT para acompañamiento técnico y simbólico

Trabajo colaborativo (reparto de tareas en el grupo de trabajo).

Ambas integrantes trabajaron por igual en el trabajo integrador, se repartieron algunas tareas. Sólo en una computadora se instaló el VirtualBox y Ubuntu Server, se realizaron videollamadas por WhatsApp y Teams durante el proceso, se compartía pantalla y debatían las distintas estrategias. En la otra computadora está la instalación del software para editar video. El trabajo fue cooperativo, integrado y muy rico en debates y aprendizajes.

5. Resultados Obtenidos

El desarrollo del trabajo permitió concretar la instalación y validación de un entorno virtual completamente funcional utilizando **VirtualBox** como hipervisor tipo 2 y **Ubuntu Server 22.04.3 LTS** como sistema operativo invitado.

A lo largo del proceso se configuraron correctamente los parámetros técnicos de la máquina virtual —asignación de memoria, espacio en disco, red y procesador—, replicando de forma precisa el comportamiento de un servidor real. Se comprobó la correcta interacción entre el hipervisor y el hardware anfitrión, garantizando la estabilidad del entorno virtual.

Durante la instalación, se empleó el instalador **Curtin**, propio de Ubuntu, y se seleccionaron cuidadosamente los **snaps** a instalar, priorizando criterios de funcionalidad, ligereza y respeto por la privacidad. Entre ellos, se incluyeron herramientas como **glances** (para monitoreo del sistema), **nextcloud** (para gestión de archivos en la nube), **nginx** (servidor web) y **postgresql** (base de datos).

También se realizaron pruebas de funcionamiento mediante comandos del sistema que permitieron comprobar el correcto desempeño del entorno.

Cada comando ejecutado fue documentado y acompañado por capturas de pantalla, lo que no solo evidencia la funcionalidad del sistema, sino también la comprensión técnica del proceso.

Se presentaron dificultades iniciales vinculadas con la **instalación del hipervisor VirtualBox** (errores de configuración, dependencias de C++ y conflictos del sistema anfitrión), que

fueron resueltas mediante la búsqueda activa de información, consultas a la documentación oficial y el apoyo de herramientas de inteligencia artificial como Copilot y ChatGPT.

Una vez superadas esas instancias, el entorno virtual respondió correctamente a todas las pruebas, mostrando estabilidad, bajo consumo de recursos y total independencia respecto del sistema anfitrión.

El resultado final es un **servidor virtual operativo**, capaz de ejecutar aplicaciones, administrar servicios y simular condiciones reales de un entorno de producción. Este logro no solo valida los objetivos técnicos del trabajo, sino que demuestra la utilidad de la virtualización como herramienta de aprendizaje, experimentación y desarrollo en programación.

6. Conclusiones

El presente trabajo integrador permitió comprender la virtualización no solo desde una perspectiva teórica, sino también práctica y reflexiva. Se comprobó que esta tecnología constituye un pilar fundamental en la arquitectura de los sistemas operativos modernos, al posibilitar la creación de entornos aislados, seguros y eficientes dentro de un mismo hardware físico.

La experiencia práctica ofreció una visión integral sobre cómo se gestiona el hardware mediante software, cómo se distribuyen los recursos virtualizados (CPU, memoria, disco, red) y cómo cada máquina virtual opera de manera autónoma sin interferir con el entorno anfitrión. A través de la configuración de **VirtualBox** y la instalación de **Ubuntu Server**, se consolidaron conocimientos sobre hipervisores, gestión de paquetes, servicios de sistema, y administración de entornos Linux.

Desde lo metodológico, el trabajo implicó una **colaboración constante** entre los integrantes, con instancias de investigación, prueba y resolución de errores. El uso de videollamadas, capturas, documentación compartida y asistencia con IA reflejó un trabajo interdisciplinario, cooperativo y dinámico.

Entre los principales aprendizajes se destacan:

- La importancia de planificar correctamente los recursos virtuales antes de crear el entorno.
- La utilidad de la virtualización como espacio seguro para probar sin comprometer el sistema real.
- El valor del software libre como herramienta ética, accesible y formativa.
- La necesidad de documentar y reflexionar cada paso, no solo desde la técnica, sino también desde la comprensión conceptual del proceso.

Las dificultades enfrentadas —errores en la instalación, dependencias no resueltas, configuraciones de red— se transformaron en oportunidades de aprendizaje, reforzando la capacidad de análisis y resolución de problemas.

Como posibles extensiones futuras, podría explorarse la automatización de tareas mediante scripts en Python dentro del entorno virtual, o la implementación de un clúster de servidores simulados para observar la comunicación entre máquinas virtuales.

En síntesis, este trabajo no solo logró su objetivo técnico, sino que también fortaleció la comprensión de los conceptos de arquitectura y sistemas operativos, evidenciando que la virtualización es una herramienta indispensable en la formación de todo técnico en programación. Permite experimentar, errar, corregir y aprender sin miedo, dentro de un entorno seguro y controlado, donde cada error se transforma en conocimiento.

7. Bibliografía

- Universidad Tecnológica Nacional. (2025). *Material de estudio: Virtualización*. Campus Virtual de la asignatura Arquitectura y Sistemas Operativos, Tecnicatura Universitaria en Programación.
- Canonical Ltd. (2024). *Ubuntu Server Documentation*. Recuperado de <https://ubuntu.com/server/docs>
- Oracle Corporation. (2024). *VirtualBox User Manual*. Recuperado de <https://www.virtualbox.org/manual/UserManual.html>
- GitHub. (2025). *Copilot: Asistente de desarrollo basado en inteligencia artificial*. Recuperado de <https://github.com/features/copilot>
- OpenAI. (2025). *ChatGPT: Asistente de inteligencia artificial para redacción y soporte técnico*. Recuperado de <https://chat.openai.com>

8. Anexos

Código Fuente en Python para el Script:

```
# Actividad. Ingresar los nombres de 3 alumnos, y para cada uno una tupla de 3
# notas.
# Luego, mostrar el promedio de cada alumno.

print("¡Hola!\n")

#Validación de que un valor sea un número

lista_completa_notas={}
promedios={}

def probar_numero(n):
    try:
        n=float(n)
        return True
    except ValueError:
        return False

#Calcular promedio de cada estudiante
def calcular_promedio(lista):
    suma=0
    for nota in lista:
        suma+=nota
    return suma / len(lista)

# Solicitar los tres nombre y notas
for i in range(3):
    #Solicitar el nombre del alumno al usuario y validar que no esté vacío
    nombre=input("¿Cuál es el nombre del estudiante que querés agregar?:
").strip().title()
    while nombre=="":
        print("\nEl nombre no puede estar vacío, vuelve a intentarlo...")
        nombre=input("¿Cuál es el nombre del estudiante que querés agregar?:
").strip().title()

    #Solicitar las notas de cada alumno y armar la tupla
    lista_notas=[]
    for j in range(1,4):
        nota=-1
        while nota<1 or nota>10:
            nota_texto=input(f"Nota {j} de {nombre}: ").strip()
```

```
        while not probar_numero(nota_texto):
            print("\nDebes ingresar un valor numérico, intenta nuevamente...")
            nota_texto=input(f"Nota {j} de {nombre}: ").strip()
            nota=float(nota_texto)
            if nota<1 or nota>10:
                print("La nota debe ser un valor entre 1 y 10, intenta nuevamente...")
            lista_notas.append(nota)
        tupla_notas=tuple(lista_notas)

        #Armar un diccionario para guardar todas las notas correspondientes a cada estudiante
        lista_completa_notas[nombre]=tupla_notas
        print(f"\nLas notas de {nombre} son {tupla_notas}\n")

        #Calcular promedios y registrarlos en un diccionario:
        promedio=calcular_promedio(lista_notas)
        print(f"El promedio de las notas de {nombre} es {promedio:.2f}\n")

        promedios[nombre]=round(promedio,2)

#Mostrar los resultados
print(f"\nLas notas registradas son:")
for alumno, nota in lista_completa_notas.items():
    print(f"{alumno}: {nota} ")
print(f"\nLos promedios registrados son:")
for alumno, media in promedios.items():
    print(f"{alumno}: {media} ")

print("\n¡Muchas gracias y hasta luego!\n")
```