



UNIVERSIDAD  
TECNOLÓGICA  
NACIONAL

# Arquitectura y Sistemas Operativos

## Trabajo Práctico Integrador

### Virtualización

Alumnos:

Matías Pérez

Marcos Pousada

Profesores:

Mauricio Gabriel Pasti

Tutor:

Nicolás Castro

# Índice

Índice.....	2
Introducción.....	3
Marco Teórico.....	4
Caso Práctico.....	6
Metodología Utilizada.....	16
Resultados Obtenidos.....	19
Conclusiones.....	19
Bibliografía.....	20
Anexos.....	20

# Introducción

En este trabajo se abordará el montaje y la configuración de un entorno virtualizado mediante VirtualBox, con Ubuntu como sistema operativo invitado, y la instalación de un servidor web Apache2 a modo de ejemplo práctico.

El tema de la virtualización ha sido elegido por la creciente relevancia que presenta en entornos de desarrollo, pruebas y producción. Al utilizar máquinas virtuales, es posible aislar proyectos, simular diversas configuraciones de software y hardware, y disponer de un entorno controlado que se puede restaurar o eliminar sin afectar al equipo físico. Para un técnico en programación, este conocimiento resulta fundamental dado que permite crear laboratorios de prueba, instalar dependencias y desplegar servicios sin comprometer la estabilidad del sistema principal.

Al integrar VirtualBox y Ubuntu, se busca alcanzar varios objetivos específicos:

1. Comprender la finalidad y los beneficios de la virtualización a nivel de software.
2. Aprender el procedimiento de creación de una máquina virtual en VirtualBox y la instalación de Ubuntu en la misma.
3. Configurar recursos (CPU, memoria, almacenamiento) de manera equilibrada para lograr un rendimiento aceptable tanto en la máquina anfitriona como en la invitada.
4. Instalar y poner en funcionamiento un servicio web básico con Apache2, de modo que sirva como demostración práctica de un entorno de servidor Linux

Con este desarrollo se busca definir las bases teóricas necesarias para que, como futuros profesionales de la programación, podamos desplegar entornos de desarrollo o prueba de manera rápida, reproducible y segura.

# Marco Teórico

## Virtualización

La virtualización se basa en crear entornos independientes de software que imitan recursos físicos (CPU, memoria, almacenamiento, red) sin necesidad de tener otro equipo. Esto permite ejecutar varios sistemas operativos al mismo tiempo en una sola computadora, cada uno aislado del otro. De esta forma, se aprovecha mejor el hardware, se prueba software sin riesgo de “romper” el equipo principal y se pueden replicar configuraciones con facilidad.

## Hipervisores

Los hipervisores son los programas que hacen posible la virtualización. Hay dos tipos principales:

- **Tipo 1 (bare-metal):** se instalan directo sobre el hardware, sin sistema operativo de por medio (por ejemplo, VMware ESXi o Xen).
- **Tipo 2 (hosted):** se instalan sobre un sistema operativo normal y luego crean las máquinas virtuales. VirtualBox entra en esta categoría, lo que lo hace más sencillo de usar si ya tenés Windows, macOS o Linux instalado.

## Oracle VirtualBox

VirtualBox es un hipervisor de tipo 2, gratuito y de código abierto, que sirve para crear y manejar máquinas virtuales en casi cualquier PC. Algunas de sus características más útiles son:

- Poder ajustar fácilmente cuánta memoria RAM, cuántos núcleos de CPU y cuánto espacio en disco se le puede asignar a cada VM.
- Sacar “instantáneas” (snapshots) de la máquina, para volver a un estado anterior si algo sale mal.
- Compartir carpetas y portapapeles entre la máquina anfitriona y la invitada, lo que facilita mover archivos sin subirlos a la nube.
- Configurar la red en modo NAT (para que la VM navegue a través del host) o en modo Puente (para que la VM tenga su propia IP en la red local).

## Ubuntu como sistema operativo invitado

Ubuntu es una de las distribuciones de Linux más populares y con gran soporte comunitario. En una VM suele funcionar bien con 2 GB de RAM y 20 GB de disco, lo que alcanza para muchos ensayos o un servidor ligero. Entre sus puntos fuertes están:

- **Gestor de paquetes APT:** permite instalar o actualizar programas de manera muy sencilla, usando comandos como `sudo apt update` y `sudo apt install <paquete>`.
- **Systemd:** el sistema que controla los servicios (daemons) en segundo plano; por ejemplo, para ver si Apache2 está activo se usa `sudo systemctl status apache2`.
- **Estructura de carpetas bien organizada:** las configuraciones están en `/etc`, los archivos web en `/var/www/html`

### Servidor web Apache2

Apache2 es uno de los servidores HTTP más extendidos en Linux. La razón principal para elegirlo en un trabajo práctico sencillo es que se instala en unos minutos y sirve como ejemplo perfecto de cómo funciona un servidor:

- Se instala con `sudo apt install apache2`.
- Al iniciarse, crea una página predeterminada que se ve en el navegador apenas abrís la IP de la VM.
- El directorio por defecto para alojar páginas web es `/var/www/html`, así que con editar un simple `index.html` ya tenés algo corriendo.
- Sirve para entender el flujo básico de un servidor Linux: servicios que arrancan en el arranque, permisos de archivos, rutas de configuración y cómo exponer un sitio a la red.

### Script en Bash

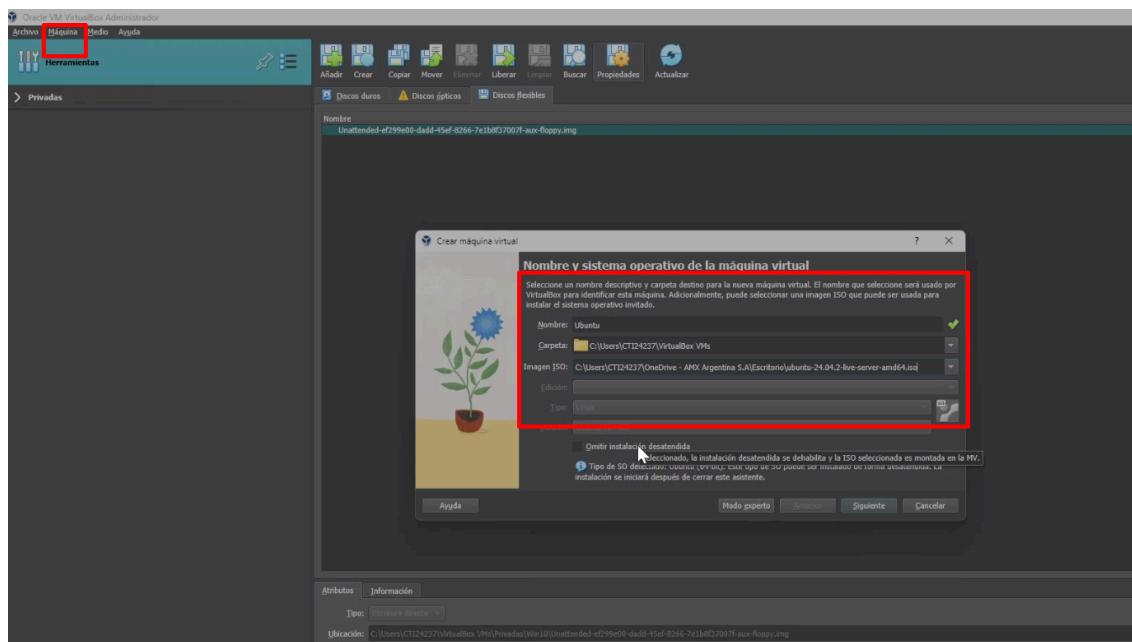
Como modo de prueba, vamos a crear un script en bash que consulte en tiempo real la información del último campeón del mundo de la copa fifa y lo guarde en un archivo html para ser servido por el servidor apache

# Caso Práctico

## Instalación de Ubuntu en Virtualbox

### Prerrequisitos:

- Equipo Anfitrión con Windows Instalado
- Virtualbox Instalado en equipo Anfitrión
- Imagen ISO descargada del Sistema operativo Invitado



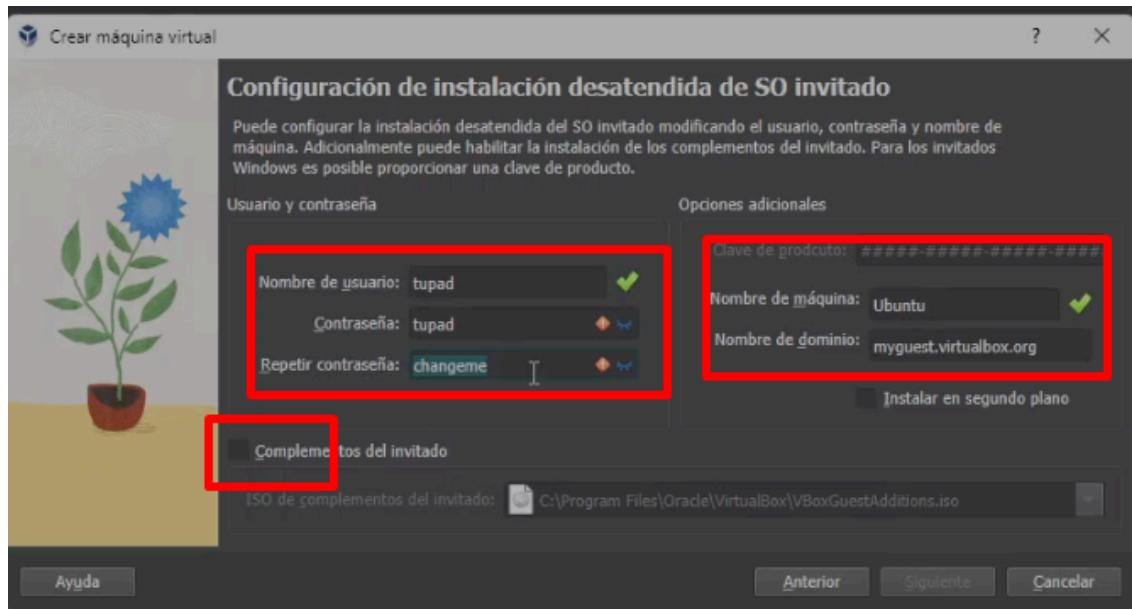
En el menú, seleccionamos la opción “Maquina” y creamos una nueva máquina virtual.

Completamos los campos a continuación:

Nombre: El nombre o alias de la máquina virtual

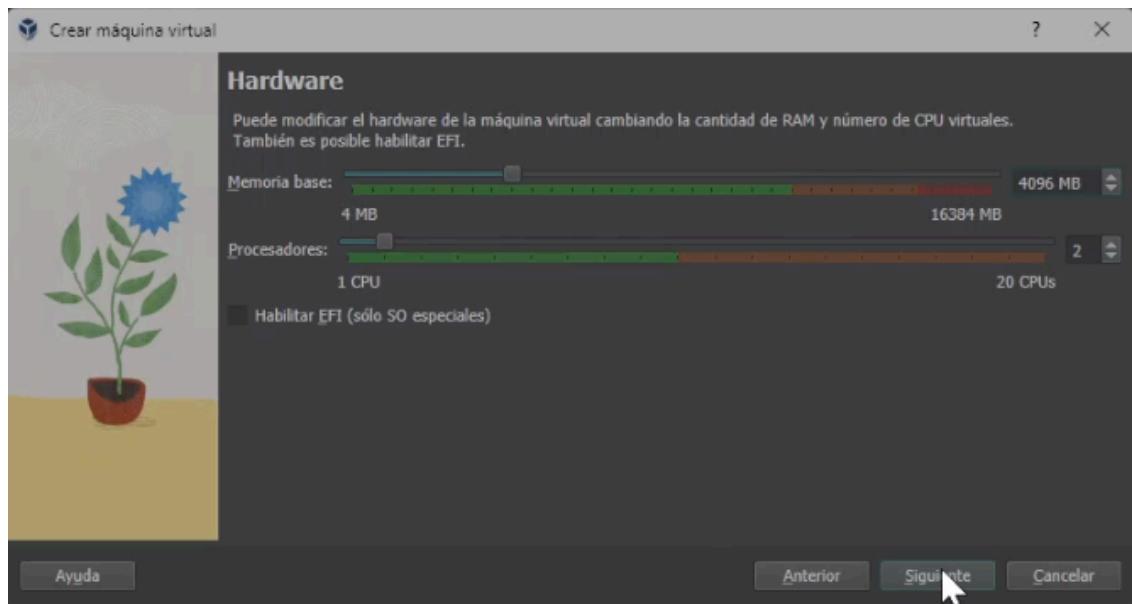
Carpetas: Directorio en donde se va a guardar la imagen de la máquina virtual

Imagen ISO: Imagen del Sistema Operativo que vamos a instalar (en este caso Ubuntu Server)



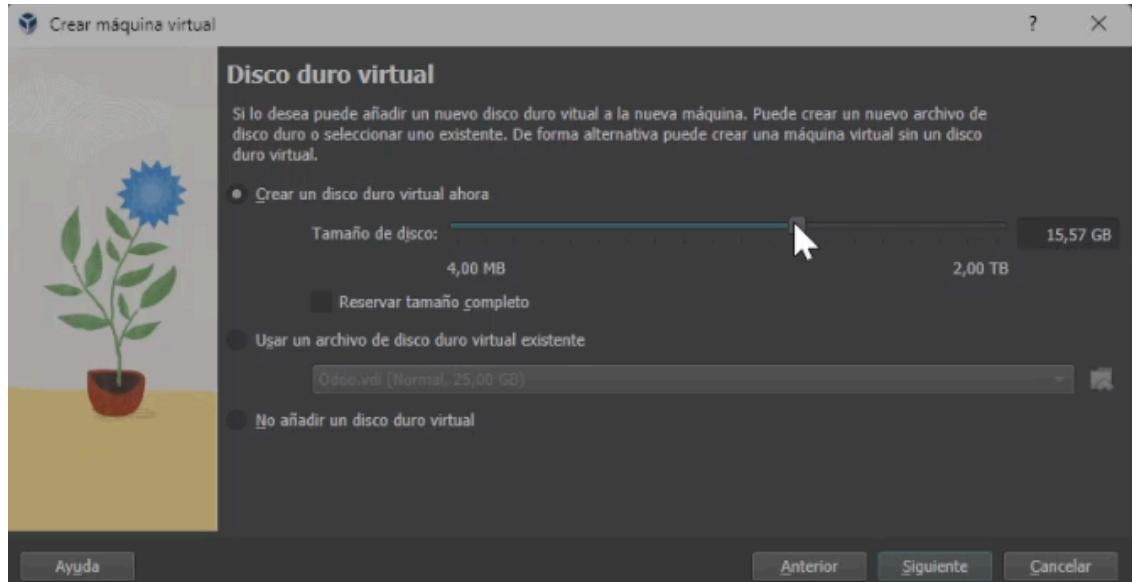
En este paso, podemos definir algunas opciones vinculadas a la instalación desatendida, nosotros vamos a definir usuario, contraseña y hostname del equipo invitado

También vamos a elegir tildar la opción “Complementos del invitado” que nos permite instalar herramientas en el sistema operativo invitado las cuales nos facilitan la interacción entre el anfitrión y el huésped

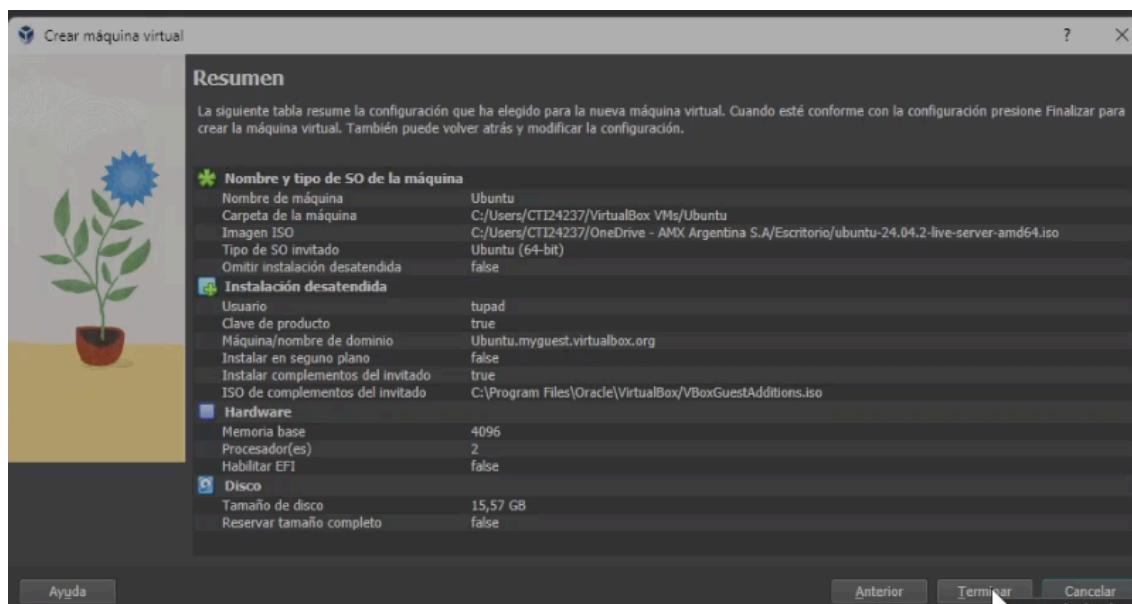


En esta pantalla se define la configuración de hardware que va a tener el equipo invitado. **Siempre en este punto se tiene que tener en cuenta al elegir esta parametrización que deben de quedar recursos disponibles en el sistema anfitrión para poder desempeñar tareas adicionales al mismo tiempo que poder ejecutar el sistema invitado**

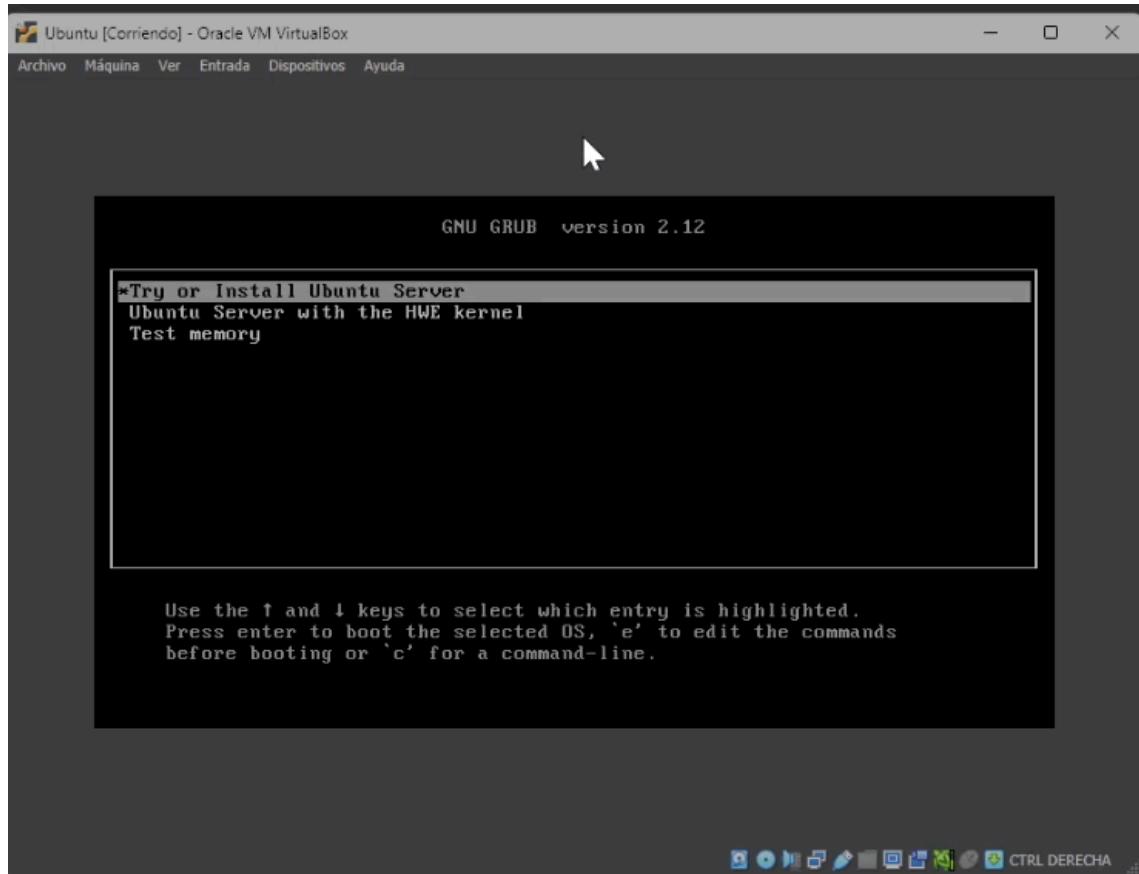
En nuestro caso de práctica vamos a configurar nuestro sistema invitado con 4gb de memoria ram y 2 cpus



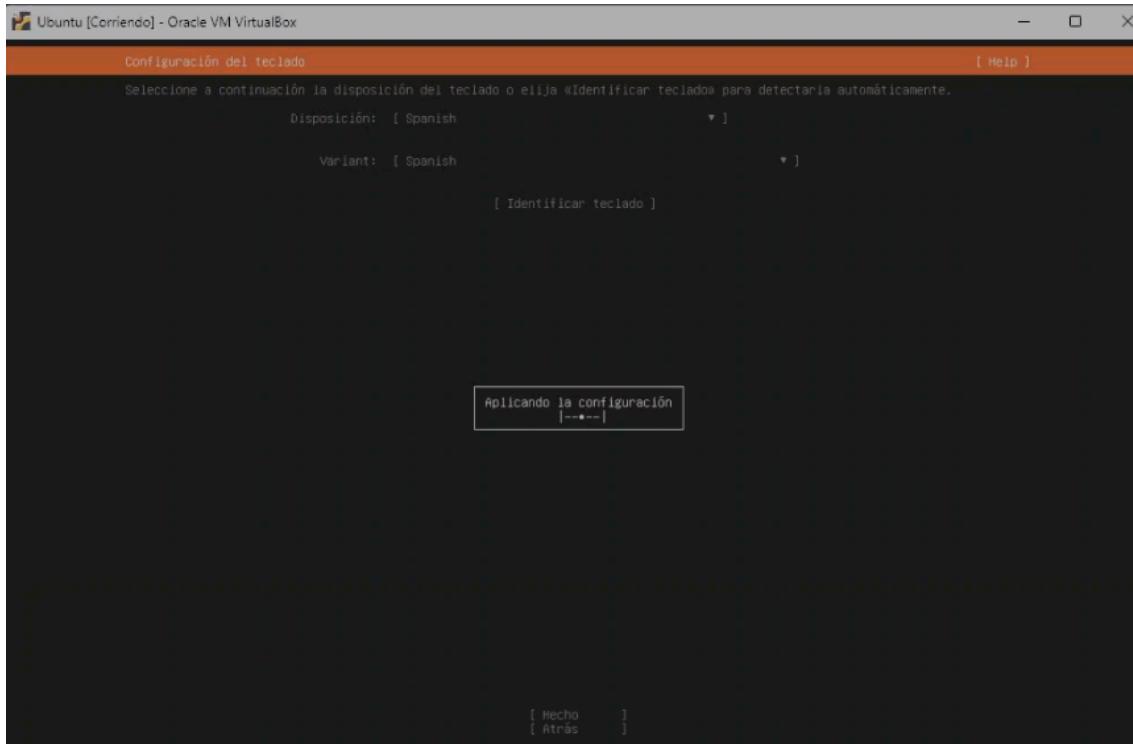
En la siguiente pantalla vamos a definir el tamaño del disco disponible para el sistema invitado, al igual que en la configuración anterior, se tiene que tener en cuenta que hay que tener espacio en disco disponible para que el sistema anfitrión funcione sin inconvenientes y para que el espacio alcance, tanto como para la instalación del sistema operativo como para la configuración de las aplicaciones que deseamos instalar en nuestro sistema invitado



En esta pantalla tenemos un resumen de las opciones elegidas durante la creación de la máquina virtual y es el paso anterior a la creación de la misma.

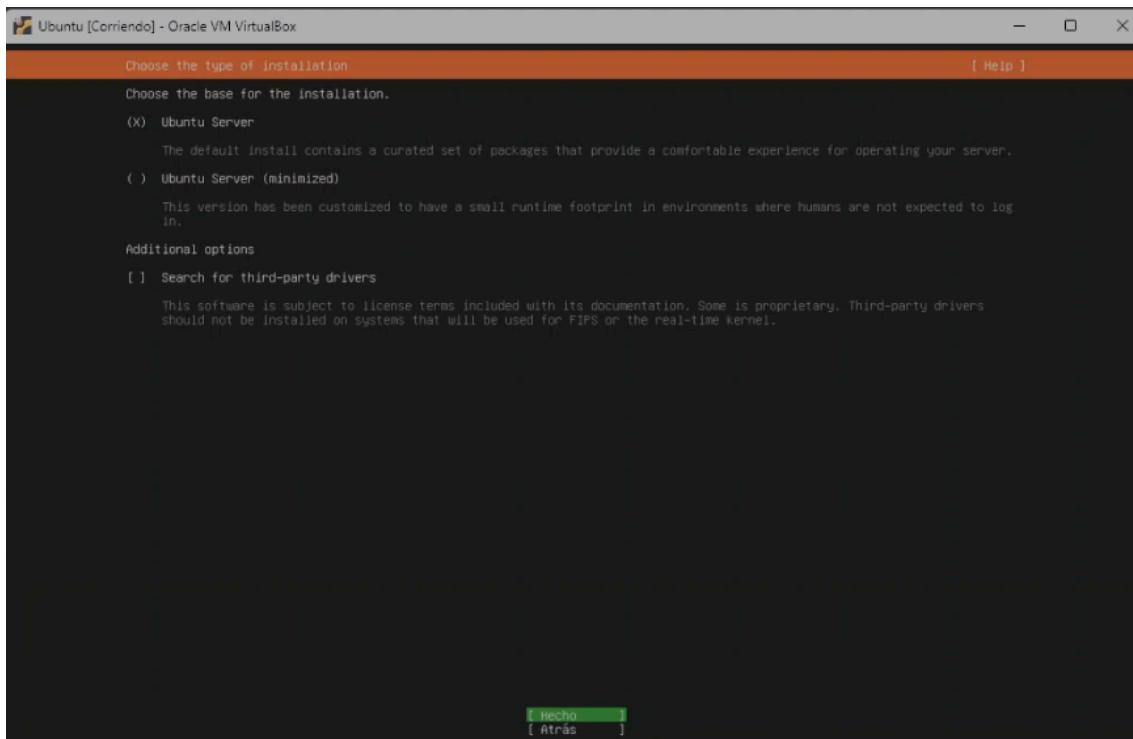


Al finalizar la creación de la máquina virtual, la misma se inicializará y la bios virtual de virtualbox va a cargar la imagen iso que configuramos en los pasos anteriores, cargando la imagen del inicio del sistema operativo que nos va a permitir instalar ubuntu.



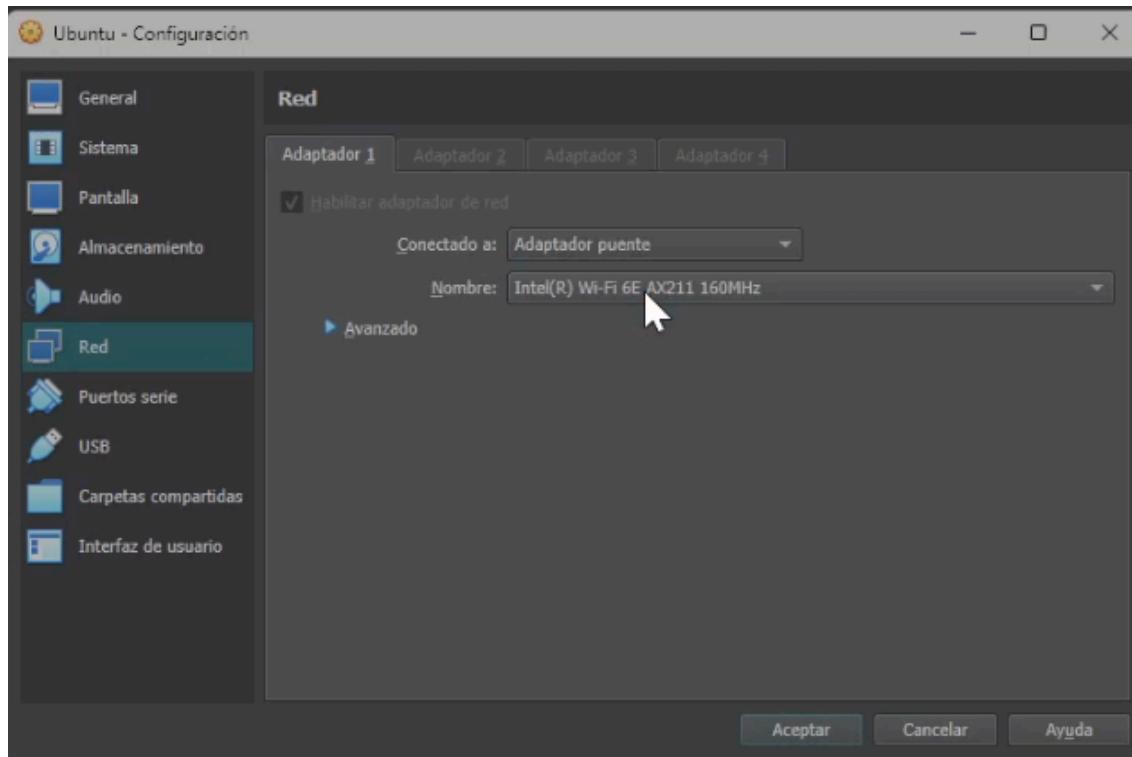
Luego de seleccionar del menú la opción de instalación vamos a pasar a las opciones de configuración de nuestro server.

En este caso elegimos el idioma y la disposición del teclado



En esta opción se puede definir el tipo de instalación, si elegimos una instalación mínima, se van a instalar una versión reducida del sistema operativo.

En nuestro caso vamos a elegir la opción por defecto.



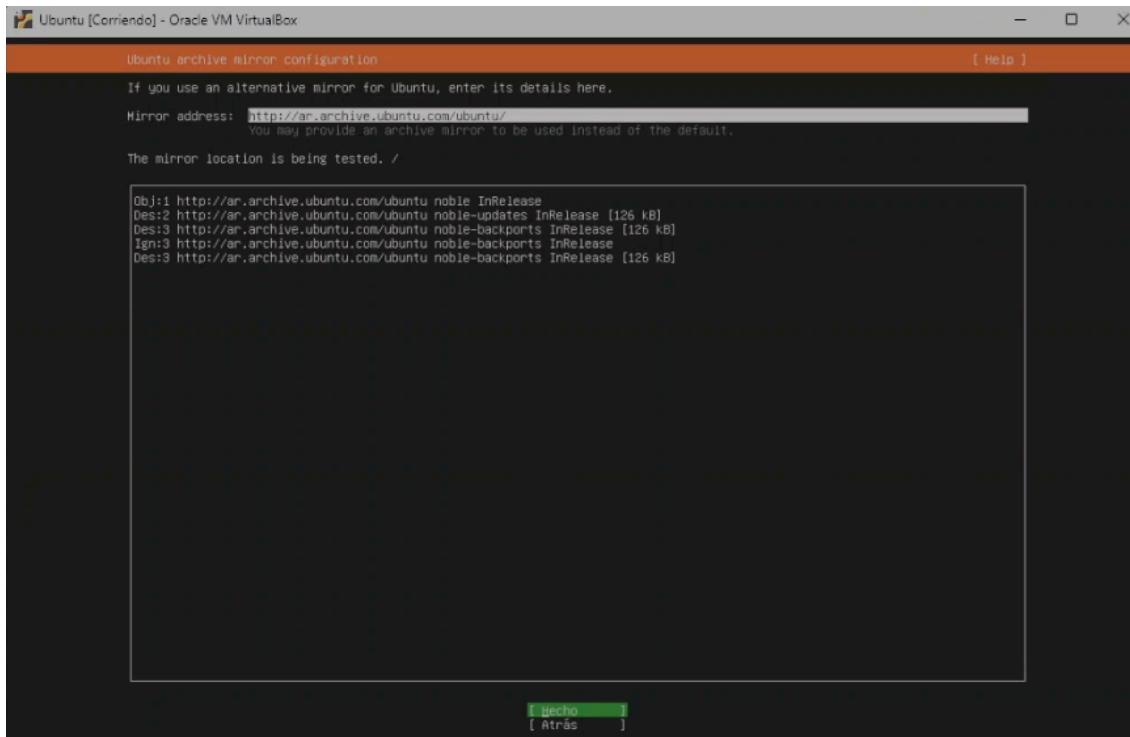
Este paso se puede configurar en cualquier punto de la instalación, dado que la configuración se puede realizar en caliente.

Es el caso de el adaptador de red. Dentro de las opciones que existen para esta configuración se pueden destacar 2

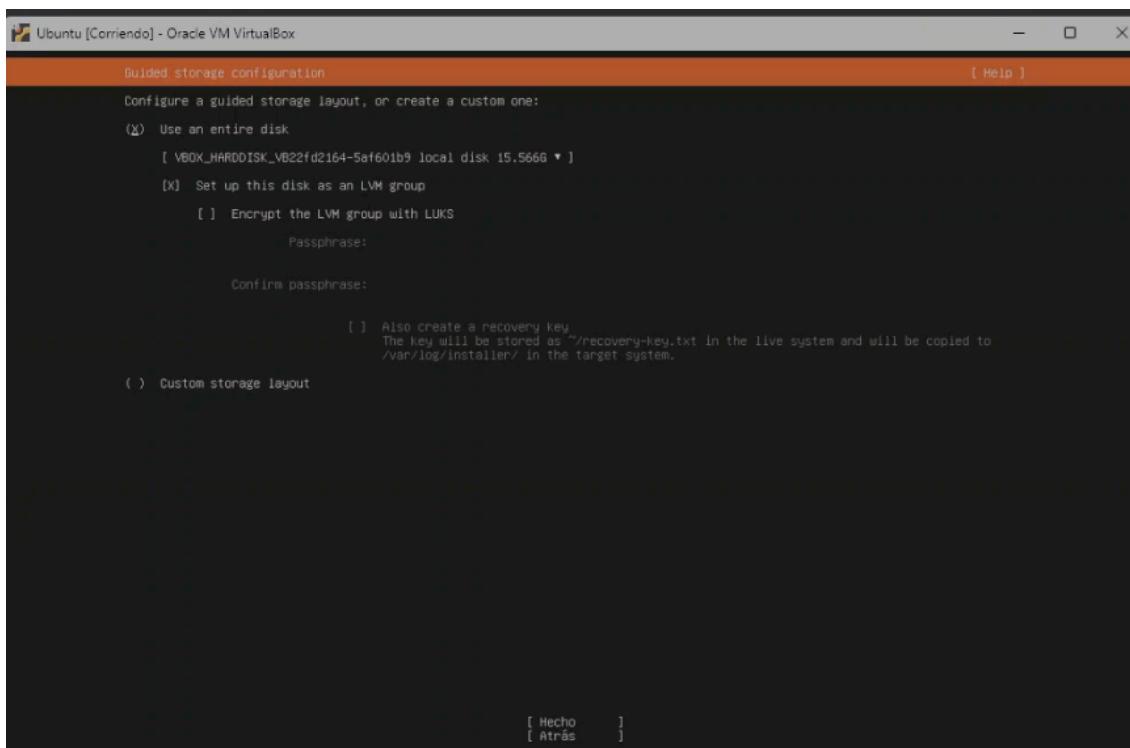
NAT: La VM accede a Internet a través del sistema anfitrión, como si estuviera "escondida" detrás de él. No es visible desde otros dispositivos de la red.

BRIDGE: La VM se conecta directamente a la red local, como si fuera otro equipo más. Tiene su propia IP y es visible desde otros dispositivos.

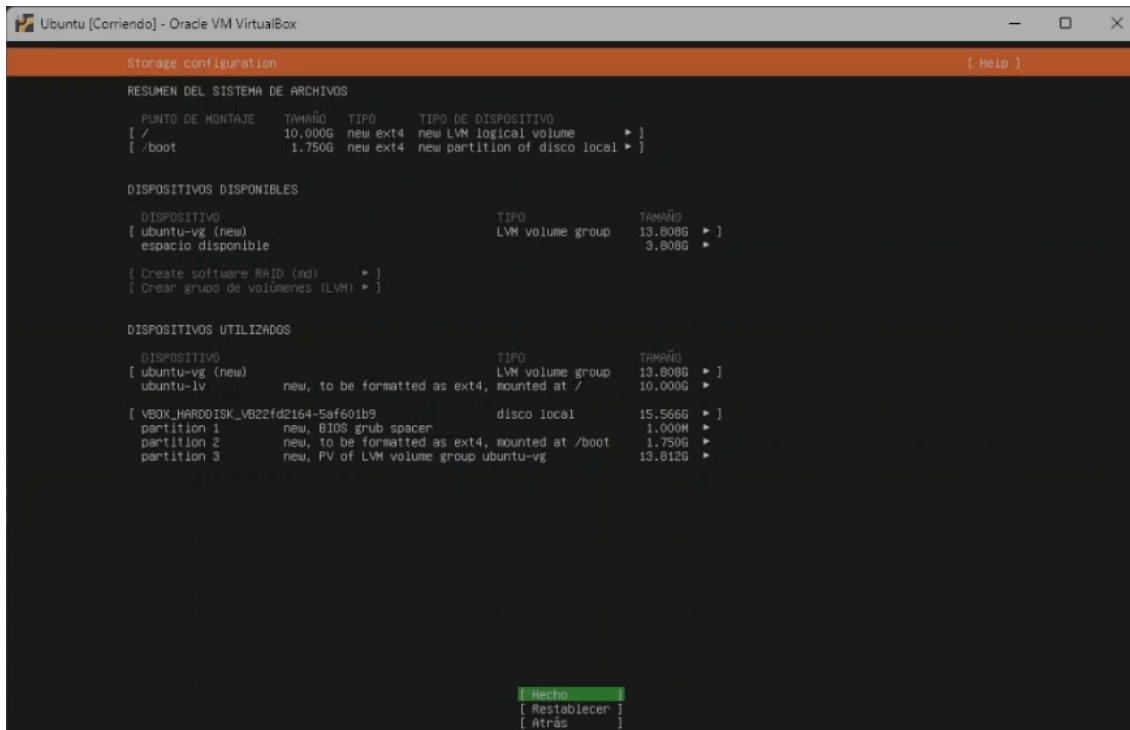
Para nuestro caso práctico, la opción recomendada es el modo Bridge



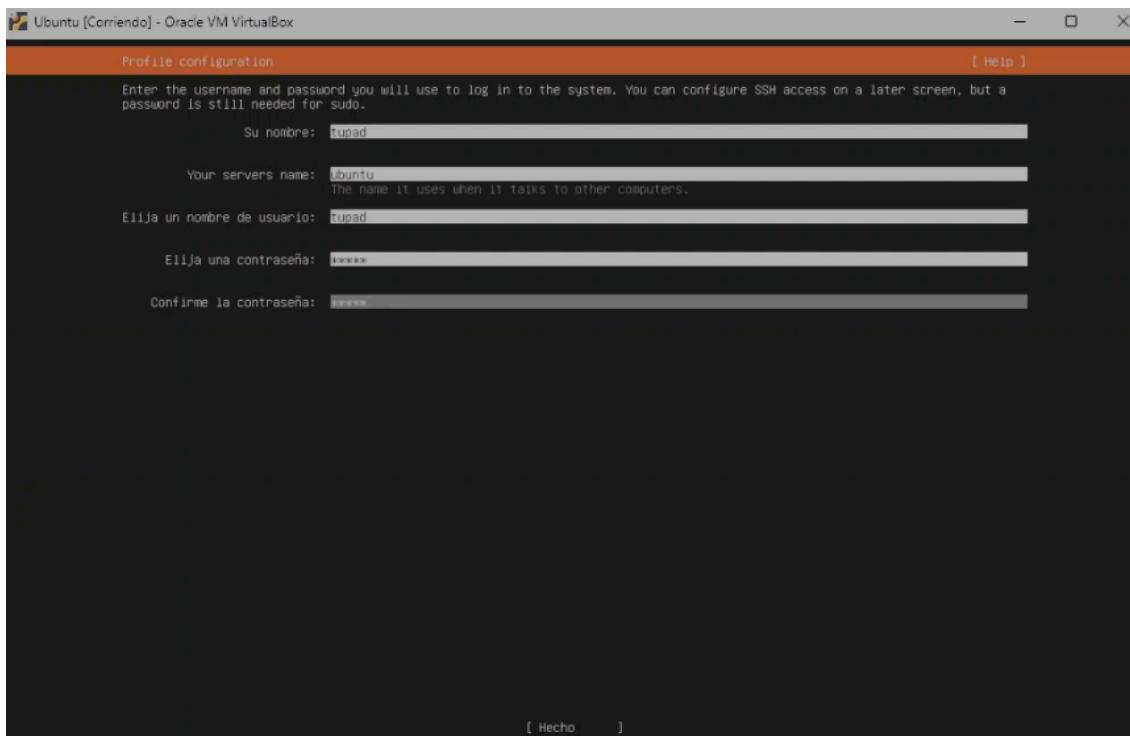
En esta etapa se configuran y actualizan los espejos de los repositorios de los paquetes del sistema operativo



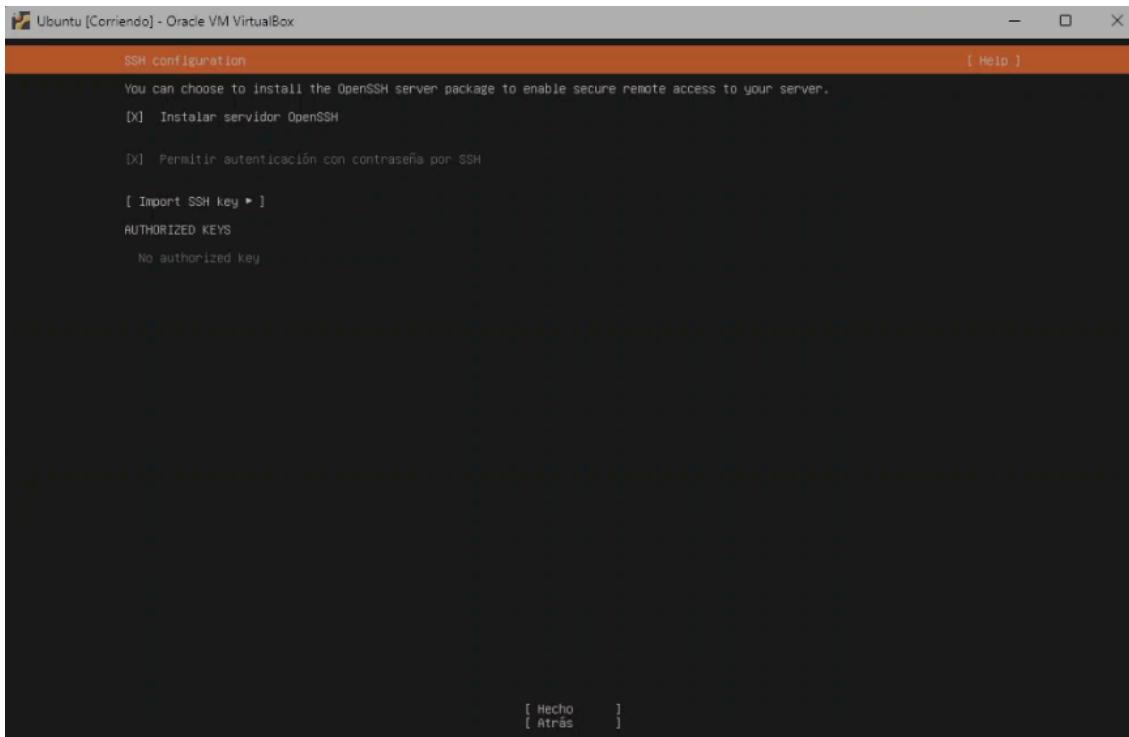
En esta pantalla se configuran parámetros relativos al uso del disco, para efectos prácticos dejaremos todo por defecto



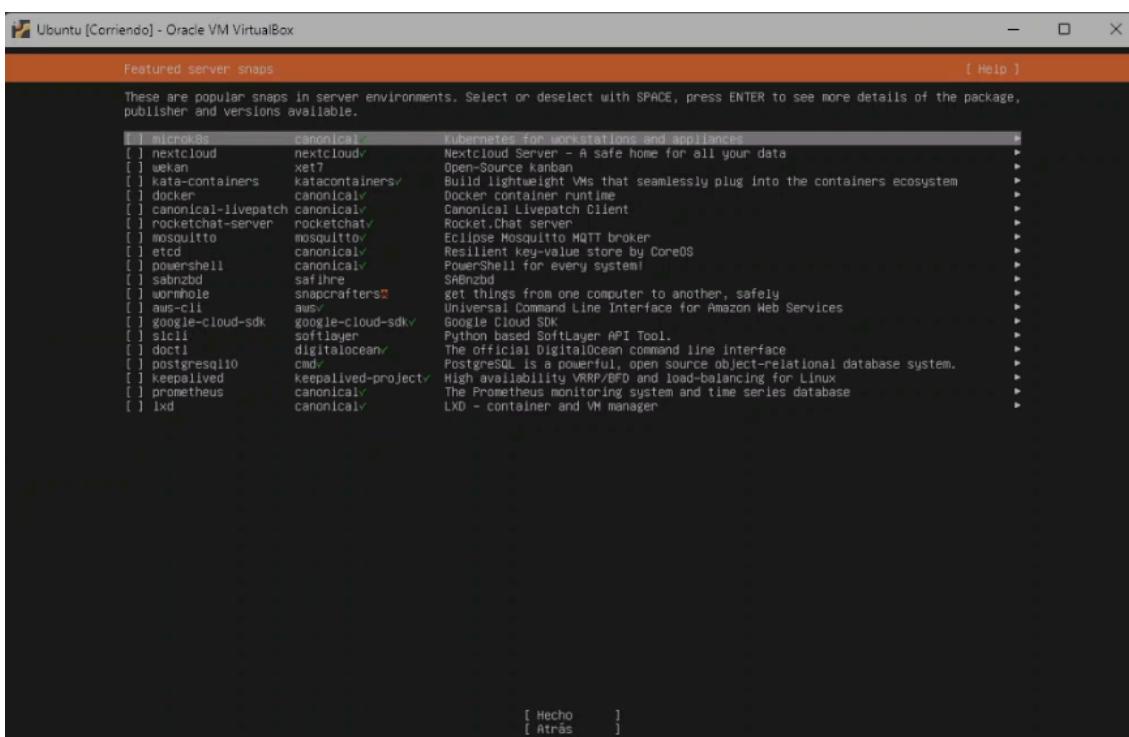
En esta pantalla se puede ver el resumen de las particiones de disco configuradas para la instalación.



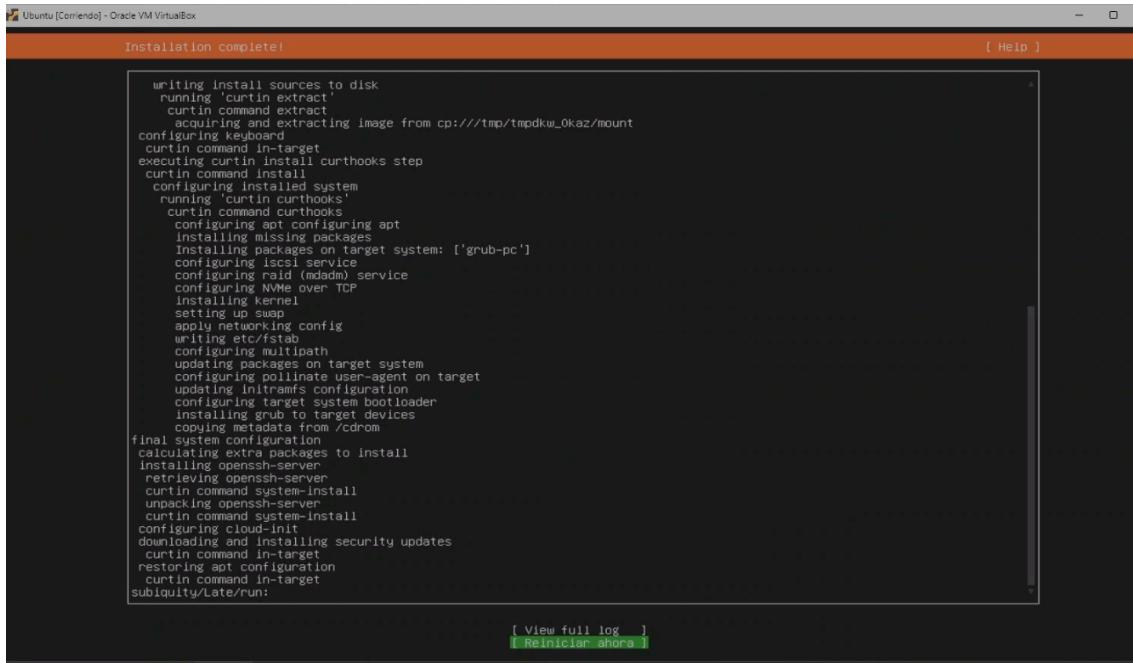
Aca definimos los valores que tendrá nuestro server como hostname user y password



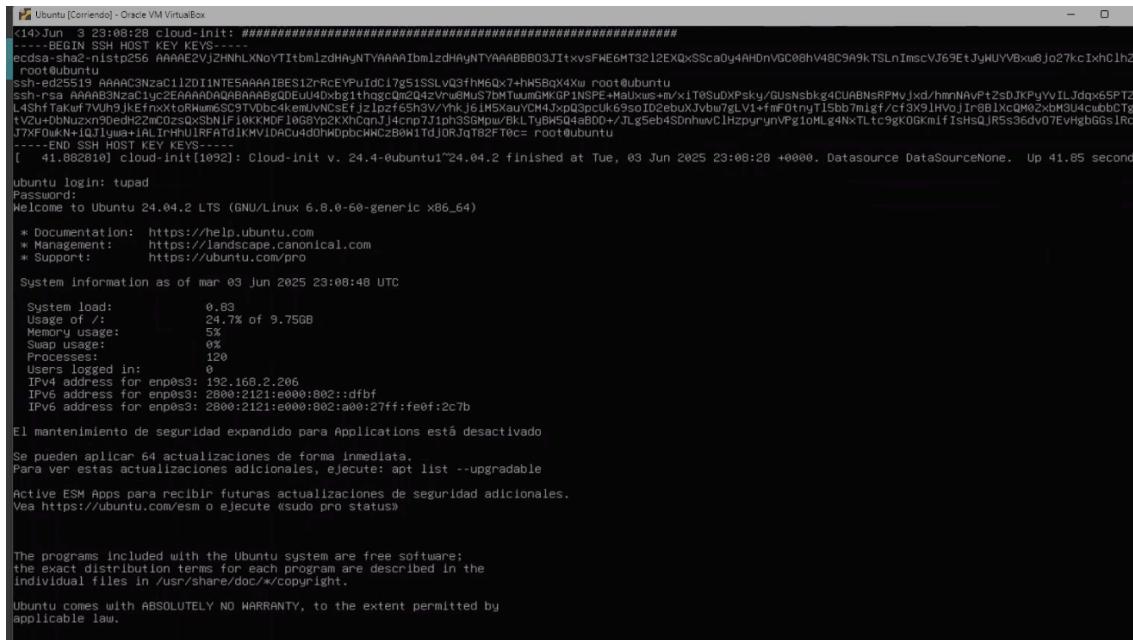
Tildamos la opción de instalar los paquetes de openSSH que más adelante nos permitirán ingresar por un cliente ssh al server



Si bien para la práctica que estamos desarrollando no serán necesarias ninguna de estas configuraciones, para otros entornos que deseemos configurar son muy útiles



Una vez completa la instalación, el sistema nos pide reiniciar



Finalizado el reinicio de nuestro servidor, podremos validar que el mismo este funcionando ingresando con el usuario y las credenciales configuradas durante la instalación.

```

Ubuntu 24.04.2 LTS ubuntu tty1
ubuntu login: tupad
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-60-generic x86_64)

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

System information as of mié 04 jun 2025 03:15:07 UTC

System load: 1.76 Processes: 111
Usage of /: 36.0% of 9.75GB Users logged in: 0
Memory usage: 5% IPv4 address for enp0s3: 192.168.0.9
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 64 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'

tupad@ubuntu:~$ python3 --version
Python 3.12.3
tupad@ubuntu:~$ sudo apt install apache2
[sudo] password for tupad:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
apache2-bin apache2-data apache2-utils libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1it64 liblubua5.4-0 ssl-cert
Paquetes sugeridos:
apache2-dbg apache2-suexec-pristine | apache2-suexec-custom www-browser
Se instalan los siguientes paquetes NUEVOS:
apache2 apache2-bin apache2-data apache2-utils libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1it64 liblubua5.4-0 ssl-cert
o actualizados: 10 nuevos se instalarán, 0 para eliminar y 65 no actualizados.
Se necesita descargar 2.084 kB de archivos.
Se utilizarán 8.094 kB de espacio de disco adicional después de esta operación.

Desea continuar? [S/n] _
```

Continuando con la configuración, verificamos que Python esté instalado, en este caso, ya contamos con la versión 3.12.3 preinstalada.

A continuación, procedemos con la instalación de Apache2. (Apache es un servidor web, es decir, un programa que se ejecuta en un computador —servidor— y permite mostrar páginas web a través de Internet o una red local).

```

Preparando para desempaquetar .../4-liblubua5.4-0.5.4.6-3build2_amd64.deb ...
Desempaquetando liblubua5.4-0:amd64 (5.4.6-3build2) ...
Seleccionando el paquete apache2-bin previamente no seleccionado.
Preparando para desempaquetar .../5-apache2-bin_2.4.58-1ubuntu0.6_amd64.deb ...
Desempaquetando apache2-bin (2.4.58-1ubuntu0.6) ...
Seleccionando el paquete apache2-data previamente no seleccionado.
Preparando para desempaquetar .../6-apache2-data_2.4.58-1ubuntu0.6_all.deb ...
Desempaquetando apache2-data (2.4.58-1ubuntu0.6) ...
Seleccionando el paquete apache2-utils previamente no seleccionado.
Preparando para desempaquetar .../7-apache2-utils_2.4.58-1ubuntu0.6_amd64.deb ...
Desempaquetando apache2-utils (2.4.58-1ubuntu0.6) ...
Seleccionando el paquete apache2 previamente no seleccionado.
Preparando para desempaquetar .../8-apache2_2.4.58-1ubuntu0.6_amd64.deb ...
Desempaquetando apache2 (2.4.58-1ubuntu0.6) ...
Seleccionando el paquete ssl-cert previamente no seleccionado.
Preparando para desempaquetar .../9-ssl-cert_1.1.2ubuntu1_all.deb ...
Desempaquetando ssl-cert (1.1.2ubuntu1) ...
Configurando ssl-cert (1.1.2ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ssl-cert.service → /usr/lib/systemd/system/ssl-cert.service.
Configurando libaprutil1:amd64 (1.7.2-3.1ubuntu0.1) ...
Configurando liblubua5.4-0:amd64 (5.4.6-3build2) ...
Configurando apache2-data (2.4.58-1ubuntu0.6) ...
Desempaquetando libaprutil1:amd64 (1.7.2-3.1ubuntu0.1) ...
Configurando libaprutil1-ldap:amd64 (1.6.3-1.1ubuntu7) ...
Configurando libaprutil1-dbd-sqlite3:amd64 (1.6.3-1.1ubuntu7) ...
Configurando libaprutil1it64:amd64 (1.6.3-1.1ubuntu7) ...
Configurando apache2-utils (2.4.58-1ubuntu0.6) ...
Configurando apache2-bin (2.4.58-1ubuntu0.6) ...
Configurando apache2 (2.4.58-1ubuntu0.6) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authn_file.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.

Progreso: [ 95%] [=====
```

Esperamos que termine la instalación de los paquetes de apache2



Una vez finalizada la instalación procedemos a crear un “script” utilizando en este caso la herramienta “nano”

```
GNU nano 7.2 campeones2.py *
import socket
import socket.getaddrinfo
import urllib.request
import re

# Forzar solo IPv4 para evitar delays
def getaddrinfo_ipv4(*args, **kwargs):
    return [ai for ai in socket.getaddrinfo(*args, **kwargs) if ai[0] == socket.AF_INET]
socket.getaddrinfo = getaddrinfo_ipv4

# Descargar HTML de Wikipedia
url = "https://en.wikipedia.org/wiki/FIFA_World_Cup"
html = urllib.request.urlopen(url).read().decode('utf-8')

# Buscar el campeón actual usando regex
match = re.search(r'Current champions.*?title="(.*?)"', html)
campeón = match.group(1).replace("_", " ") if match else "Desconocido"

# Imagen fija de Messi
imagen_url = "https://img.goodfon.com/original/1920x1080/a/95/lionel-messi-fifa-world-cup-2022-soccer-trophy-argentina-vic.jpg"

# Generar HTML
html_salida = f"""
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Campeón del Mundo</title>
</head>
<body style="text-align:center; font-family:sans-serif;">
<h1>Último Campeón del Mundo</h1>
<p style="font-size:24px;">(campeón)</p>

</body>
</html>
"""

# Guardar en la carpeta de Apache
with open("/var/www/html/index.html", "w") as f:
    f.write(html_salida)

[Help Exit] [Write Out Read File] [Where Is Replace] [Cut Paste] [Execute Justify] [Location Go To Line] [Undo Redo] [Set Mark Copy] [To Bracket Where Was] [Previous Next]
```

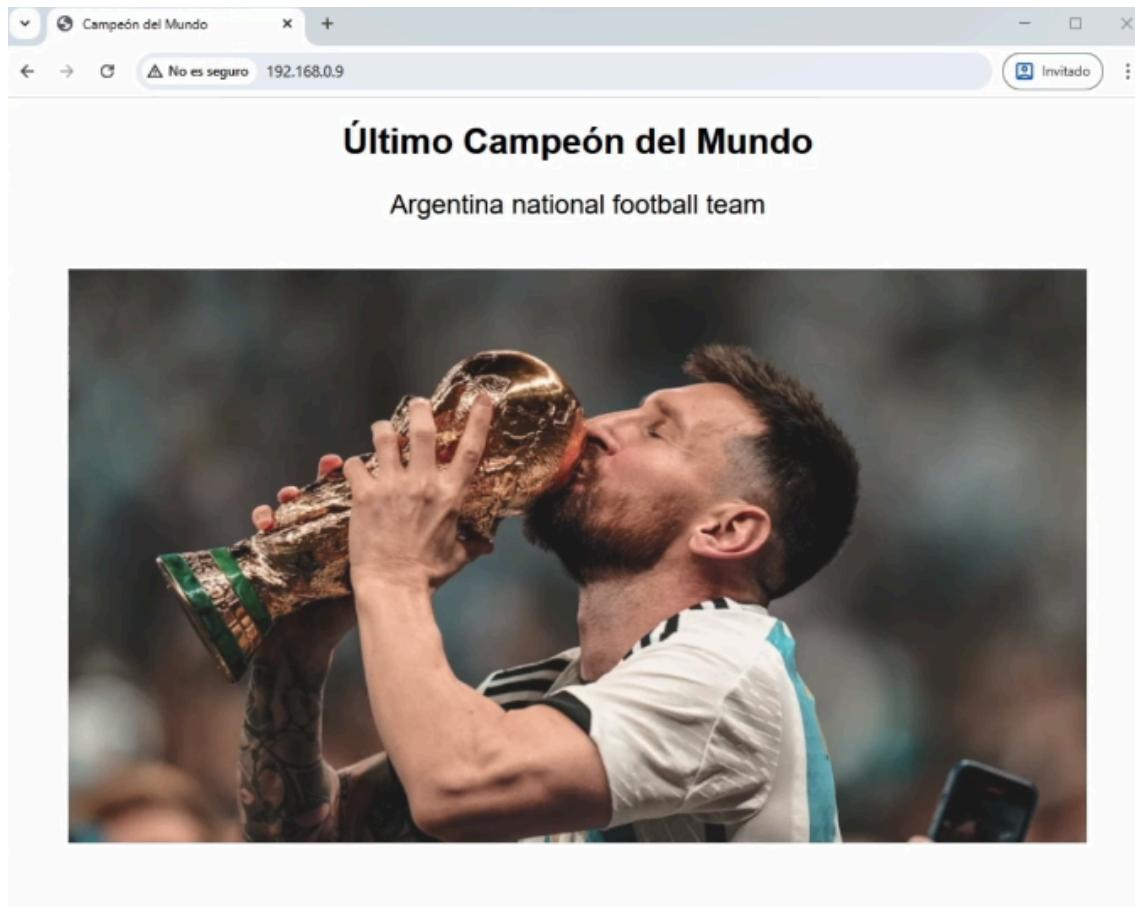
En este caso se desarrolló una script en código python que genera una página web.



Inicializamos el servidor web Apache2 en el sistema.



Ejecutamos la script campeon2.py



Finalmente, comprobamos que Ubuntu Server se está ejecutando correctamente en VirtualBox y que el script desarrollado funciona como se espera, utilizando un Navegador para hacer la prueba.

# Metodología Utilizada

Se realizó la instalación de virtualbox dentro de un host Windows utilizando como guest OS Ubuntu Server

Se instaló Apache 2 como Web server para hospedar una página web generada mediante un script en bash

En el transcurso de la práctica, se utilizaron las tecnologías de Virtualización con hypervisor, instalación y configuración de un Servidor de aplicaciones

# Resultados Obtenidos

- Se instaló el hypervisor sin inconvenientes
- Se Instaló un servidor Linux
- Se Instaló y configuró un web server
- Se desarrolló y desplegó un script que genera una página web

# Conclusiones

Este trabajo permitió entender cómo funciona la virtualización y por qué es una herramienta útil en el campo de la programación. Las aplicaciones elegidas para el trabajo ayudaron a practicar la instalación y configuración de un sistema operativo en un entorno aislado

La instalación de Apache sirvió como un ejemplo claro de cómo se puede configurar un servicio básico en linux para poder desplegar nuestro código y que sea rápidamente accesible desde la red

Las prácticas nos brindaron las herramientas necesarias para poder entender cómo configurar un entorno de virtual con muchísimas herramientas, sin poner en riesgo la máquina huésped, pudiendo reutilizar la máquina virtual tantas veces como sea necesario, clave para la formación técnica y para adquirir experiencia en el manejo de entornos de prueba y despliegue

# Bibliografía

- <https://www.virtualbox.org/wiki/Documentation>
- <https://help.ubuntu.com/stable/ubuntu-help/index.html>
- <https://httpd.apache.org/docs/2.4/>

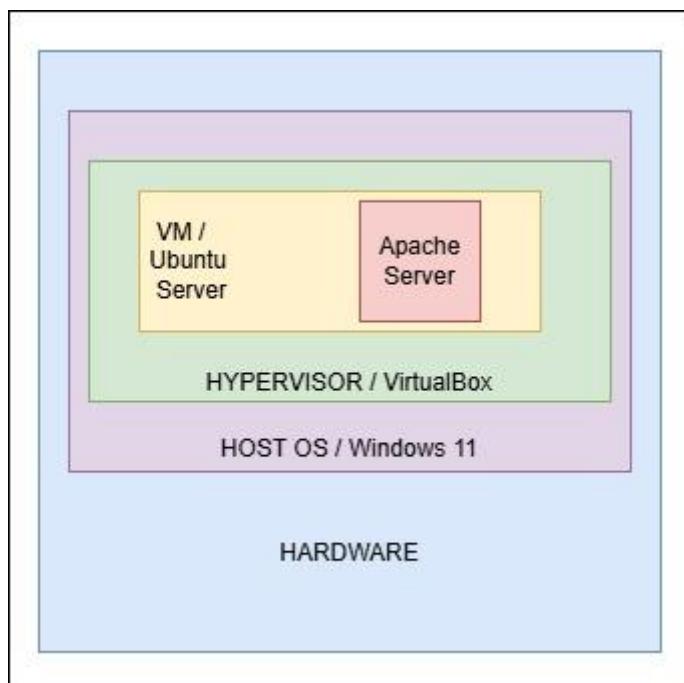
# Anexos

**Link al video explicativo:**

[https://www.youtube.com/watch?v=NPsAEVxCGkg&ab\\_channel=MatiasPerez](https://www.youtube.com/watch?v=NPsAEVxCGkg&ab_channel=MatiasPerez)

**Link del repositorio github:**

<https://github.com/matiasplm/Trabajo-Practico-AySO.git>



**Diagrama de Arquitectura de Hypervisor Tipo 2**

