

Trabajo Práctico

Virtualización

Alumnos:

- Alderete Daniel(danielalderete513@gmail.com)
- Acosta Tadeo (tadeoacosta14@gmail.com)

Materia: Arquitectura y Sistemas Operativos

Comisión: 6

Profesores: Diego Lobos – Carlos Odiard

Fecha de entrega: 05/06/2025

Índice

1. Introducción	pág. 3
2. Marco Teórico	pág. 3
3. Caso Práctico	pág. 7
4. Metodología Utilizada	pág.15
5. Resultados Obtenidos	pág.16
6. Conclusiones	pág. 17
7. Bibliografía	pág. 17
8. Anexos	pág. 17

1. Introducción:

La virtualización es una tecnología fundamental en el ámbito de la informática y desempeña un papel crucial en el desarrollo de entornos flexibles, escalables y eficientes. Elegimos este tema (virtualización) debido a su creciente relevancia y su impacto en el mundo de la programación, donde la capacidad de gestionar recursos virtualizados es esencial para la optimización de servidores y aplicaciones.

Como futuros técnicos en programación, comprender la virtualización nos permite adquirir habilidades clave en la gestión de infraestructuras, el despliegue de entornos de desarrollo y la administración eficiente de recursos informáticos. Además, nos proporciona una base sólida para trabajar con tecnologías como la computación en la nube y los contenedores, elementos fundamentales en la actualidad.

Durante el recorrido de este trabajo, buscaremos alcanzar diversos objetivos, entre los que se destacan: la comprensión de los principios de virtualización, el análisis de sus beneficios y desafíos, y la aplicación práctica de herramientas de virtualización en entornos de desarrollo.

2. Marco Teórico:

- **¿Qué es la virtualización?**

La virtualización es un proceso que permite a una computadora compartir sus recursos de hardware con varios entornos separados de forma digital. Cada entorno virtualizado se ejecuta dentro de los recursos asignados, como la memoria, la potencia de procesamiento y el almacenamiento. Con la virtualización, las organizaciones pueden cambiar entre diferentes sistemas operativos en el mismo servidor sin tener que reiniciar.

Las máquinas virtuales y los hipervisores son dos conceptos importantes en la virtualización. A continuación, mostraremos una breve definición:

Máquina virtual:

Una *máquina virtual* es un equipo definido por software que se ejecuta en un equipo físico con un sistema operativo y recursos informáticos independientes. La computadora física se denomina *máquina host* y las máquinas virtuales son *máquinas invitadas*. Se pueden ejecutar varias máquinas virtuales en una sola

máquina física. Un hipervisor extrae las máquinas virtuales del hardware de la computadora.

Hipervisor:

El *hipervisor* es un componente de software que administra varias máquinas virtuales en una computadora. Garantiza que cada máquina virtual reciba los recursos asignados y no interfiera con el funcionamiento de otras máquinas virtuales.

- **¿Cuáles son los beneficios de la virtualización?**

Utilización eficiente de los recursos:

La virtualización mejora los recursos de *hardware* que se utilizan en el centro de datos. Por ejemplo, en lugar de ejecutar un servidor en un sistema informático, se puede crear un grupo de servidores virtuales en el mismo sistema informático, al utilizar y devolver servidores al grupo según sea necesario. Tener menos servidores físicos subyacentes libera espacio en el centro de datos y supone un ahorro de dinero en electricidad, generadores y aparatos de refrigeración.

Administración automatizada de las TI:

Ahora que las computadoras físicas son virtuales, se pueden administrar mediante el uso de herramientas de *software*. Los administradores crean programas de implementación y configuración para definir plantillas de máquinas virtuales. Es posible duplicar la infraestructura de forma repetida y coherente y evitar las configuraciones manuales propensas a errores.

Recuperación de desastres más rápida:

Cuando eventos como los desastres naturales o los ataques cibernéticos afectan negativamente a las operaciones empresariales, recuperar el acceso a la infraestructura de TI y sustituir o arreglar un servidor físico puede llevar horas o incluso días. Por el contrario, al utilizar entornos virtualizados, el proceso tarda minutos. Esta rápida respuesta mejora significativamente la capacidad de recuperación y facilita la [continuidad del negocio](#) para que las operaciones puedan continuar según lo previsto.

- **Desafíos que presenta la virtualización:**

En la era de la tecnología y la información, las organizaciones están adoptando cada vez más la virtualización como una solución para mejorar la eficiencia y la disponibilidad de sus sistemas. La virtualización permite crear múltiples máquinas virtuales en un solo servidor físico, lo que ahorra costos, reduce el espacio físico necesario y mejora la escalabilidad. Sin embargo, cuando se trata de entornos de alta disponibilidad, la virtualización puede plantear varios desafíos. Algunos de ellos pueden ser:

1. **Rendimiento:**

Uno de los desafíos más comunes al virtualizar entornos de alta disponibilidad es el rendimiento. La virtualización agrega una capa adicional de complejidad y consumo de recursos. Cada máquina virtual tiene sus necesidades de recursos, como CPU, memoria y almacenamiento. Si no se asignan adecuadamente los recursos, las máquinas virtuales pueden verse afectadas por una falta de rendimiento, lo que puede conducir a tiempos de respuesta lentos y una experiencia deficiente para los usuarios finales. Para abordar este desafío, es importante realizar un monitoreo constante del rendimiento de las máquinas virtuales y realizar ajustes en la asignación de recursos según sea necesario.

2. **Gestión de almacenamiento:**

Otro desafío que surge al virtualizar entornos de alta disponibilidad es la gestión de almacenamiento. La virtualización puede resultar en una mayor demanda de almacenamiento, ya que cada máquina virtual requiere su propio espacio de disco duro virtual. Esto puede llevar a problemas de capacidad y dificultades para administrar el almacenamiento de manera eficiente. Una solución para este desafío es implementar tecnologías de almacenamiento compartido, como el almacenamiento en red (NAS) o el almacenamiento de área de almacenamiento en red (SAN). Estas tecnologías permiten centralizar y compartir el almacenamiento entre múltiples máquinas virtuales, lo que facilita la gestión y el escalado del almacenamiento.

3. **Conectividad de red:**

La conectividad de red es otro desafío crítico que enfrentan los entornos de alta disponibilidad virtualizados. El tráfico de red de las máquinas virtuales debe ser gestionado de manera eficiente para

garantizar un rendimiento óptimo y una disponibilidad continua. Los problemas de conectividad de red pueden surgir debido a una infraestructura de red inadecuada, cuellos de botella en el ancho de banda y problemas de configuración. Es fundamental contar con una infraestructura de red sólida y confiable, así como monitorear el tráfico de red y realizar ajustes según sea necesario.

4. Recuperación ante desastres:

La recuperación ante desastres es un componente clave de los entornos de alta disponibilidad. En caso de fallos o interrupciones, es fundamental tener un plan de recuperación bien definido y probarlo regularmente para garantizar una recuperación rápida y sin problemas. Al virtualizar entornos de alta disponibilidad, es importante considerar cómo se realizará la recuperación de las máquinas virtuales en caso de un desastre. Esto puede implicar la replicación de máquinas virtuales en ubicaciones geográficas diferentes o la implementación de soluciones de respaldo y recuperación en la nube.

5. Gestionar la complejidad:

Finalmente, uno de los desafíos más importantes para la virtualización en entornos de alta disponibilidad es la gestión de la complejidad. La virtualización agrega una capa adicional de complejidad a la infraestructura de TI, lo que puede dificultar la gestión y el mantenimiento del sistema. La gestión de múltiples máquinas virtuales, la asignación de recursos, la configuración de la red y la seguridad pueden volverse abrumadoras si no se planifican y gestionan adecuadamente. Para superar este desafío, es esencial implementar herramientas de gestión y monitoreo adecuadas, así como contar con personal capacitado en virtualización y alta disponibilidad.

- **¿Cómo funciona la virtualización?**

La virtualización utiliza un software especializado, llamado hipervisor, para crear varias instancias en la nube o máquinas virtuales en un solo equipo físico.

Instancias en la nube o máquinas virtuales:

Después de instalar el *software* de virtualización en la computadora, podrá crear una o más máquinas virtuales. Se puede acceder a las máquinas virtuales de la misma manera que se accede a otras aplicaciones en la computadora. La computadora se llama *host* y la máquina virtual se llama huésped. Varios huéspedes se pueden ejecutar en el *host*. Cada huésped tiene su propio sistema operativo, que puede ser el mismo o diferente del sistema operativo del *host*.

Desde la perspectiva del usuario, la máquina virtual funciona como un servidor típico. Tiene ajustes, configuraciones y aplicaciones instaladas. Los recursos de computación, como las unidades centrales de procesamiento (CPU), la memoria de acceso aleatorio (RAM) y el almacenamiento aparecen de la misma manera que en un servidor físico. También es posible configurar y actualizar los sistemas operativos huéspedes y sus aplicaciones según sea necesario sin afectar al sistema operativo *host*.

Hipervisores:

El hipervisor es el *software* de virtualización que se instala en la máquina física. Es una capa de *software* que actúa como intermediario entre las máquinas virtuales y el *hardware* subyacente o el sistema operativo del *host*. El hipervisor coordina el acceso al entorno físico de manera que varias máquinas virtuales tengan acceso a su propia cuota de recursos físicos.

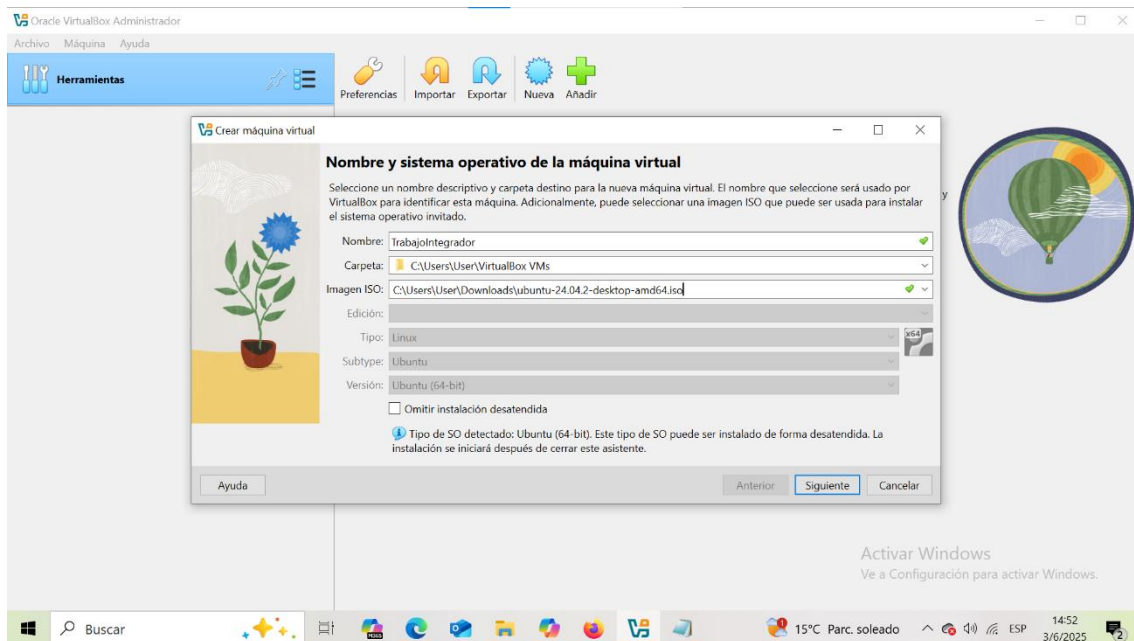
Por ejemplo, si la máquina virtual requiere recursos de computación, como potencia de procesamiento de la computadora, la solicitud se dirige primero al hipervisor. El hipervisor transmite entonces la solicitud al *hardware* subyacente, que realiza la tarea.

3. Caso Práctico:

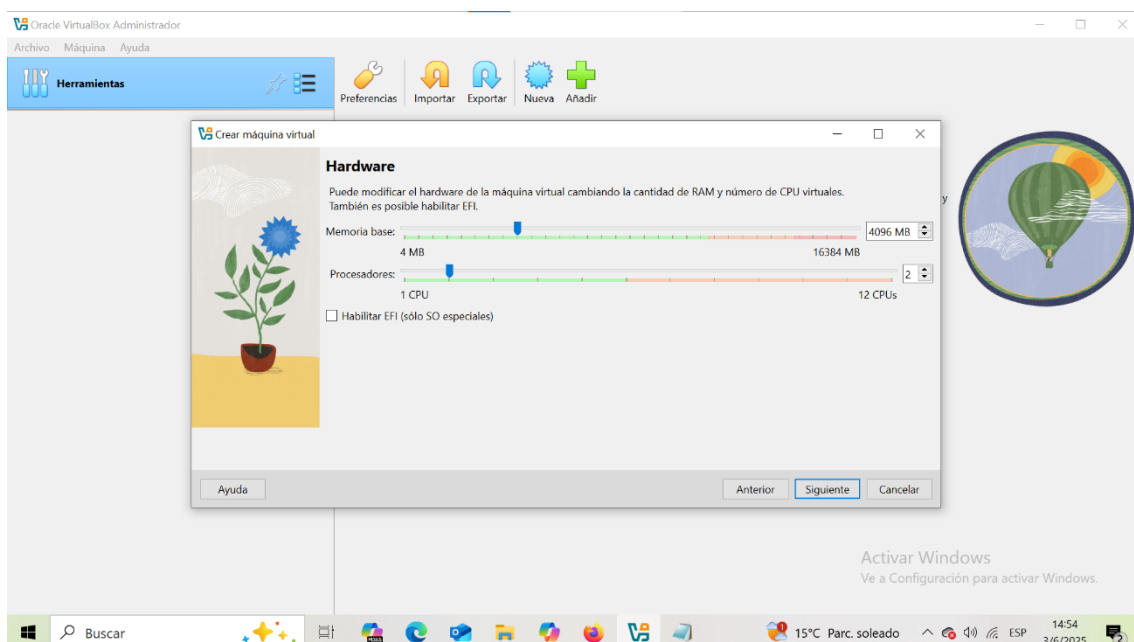
Este ejercicio abarca la **instalación de Apache**, la **configuración de la red**, el **ajuste del firewall** y la **prueba de conexión desde el navegador**.

1. Creación de una Máquina virtual en VirtualBox:

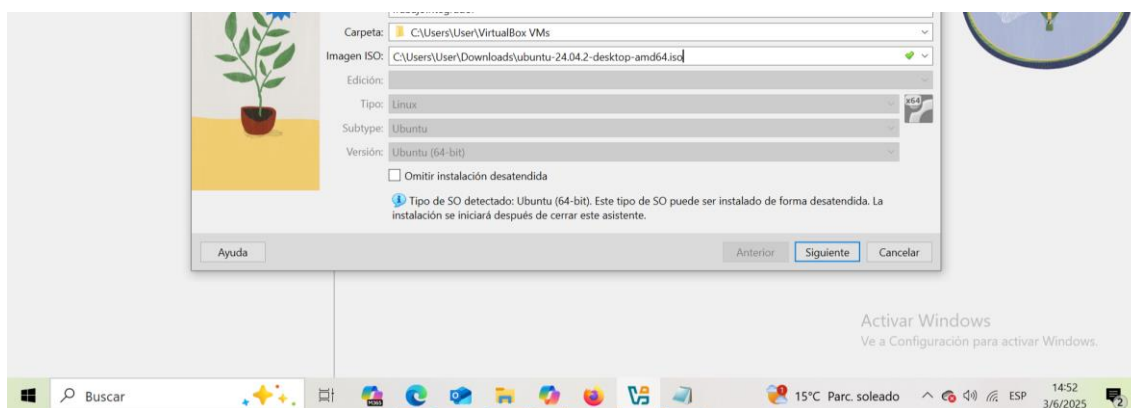
A. Abrir VirtualBox y crear una máquina virtual.



B. Asignarle 4 GB de RAM y 30 GB disco duro.

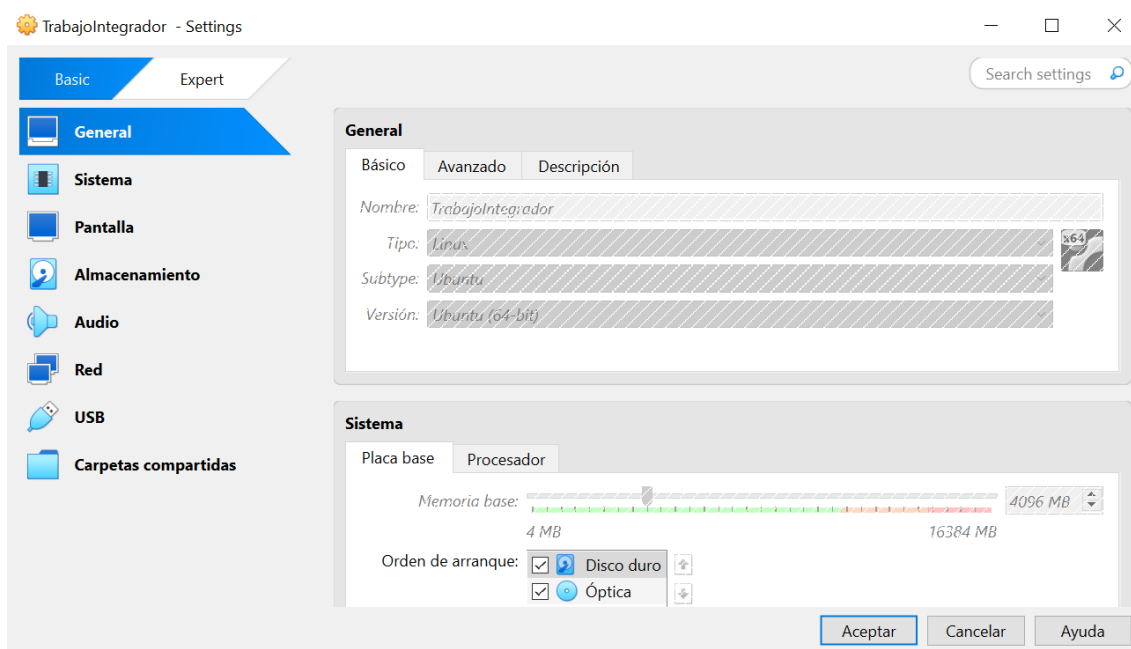


C. Seleccionar la ISO de Ubuntu Server 22.04 y utilizarlo como sistema operativo.

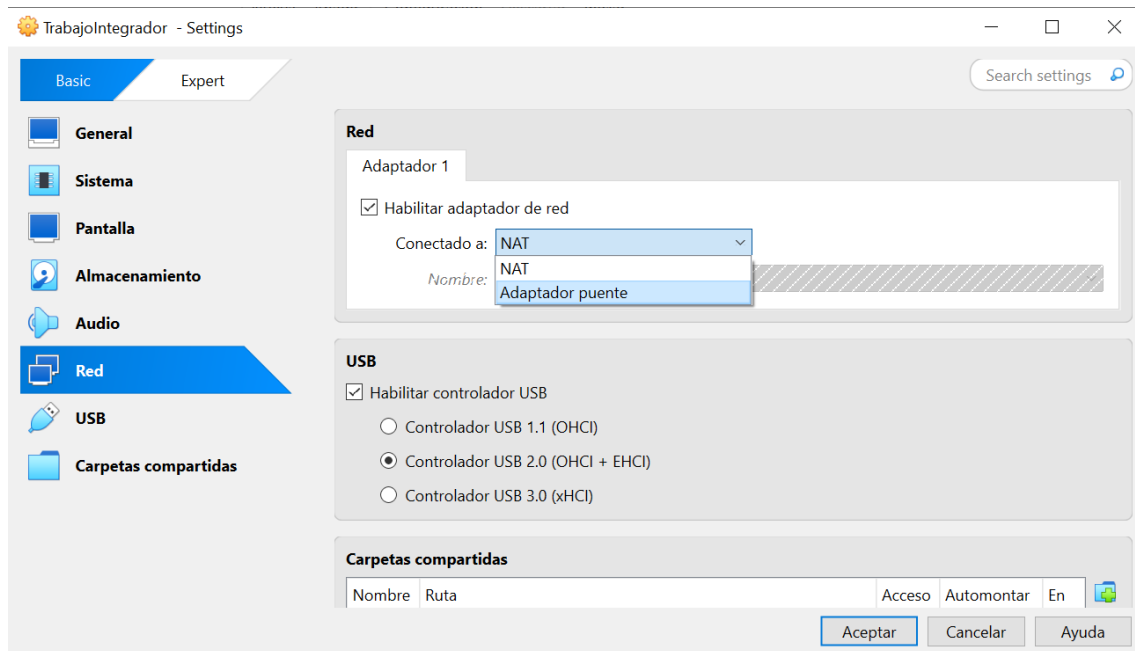


2. Configuración de red en la Máquina virtual:

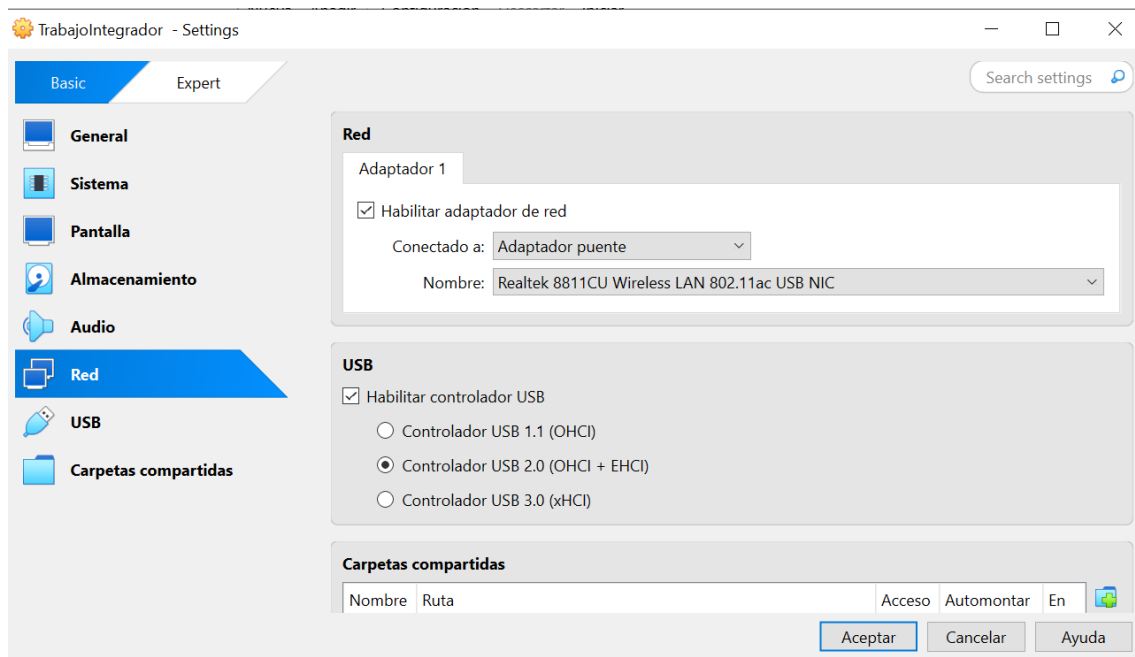
A. Ir a Configuración → Red en VirtualBox.



B. Cambiar el modo de red a "Adaptador puente" para permitir acceso desde la red local.



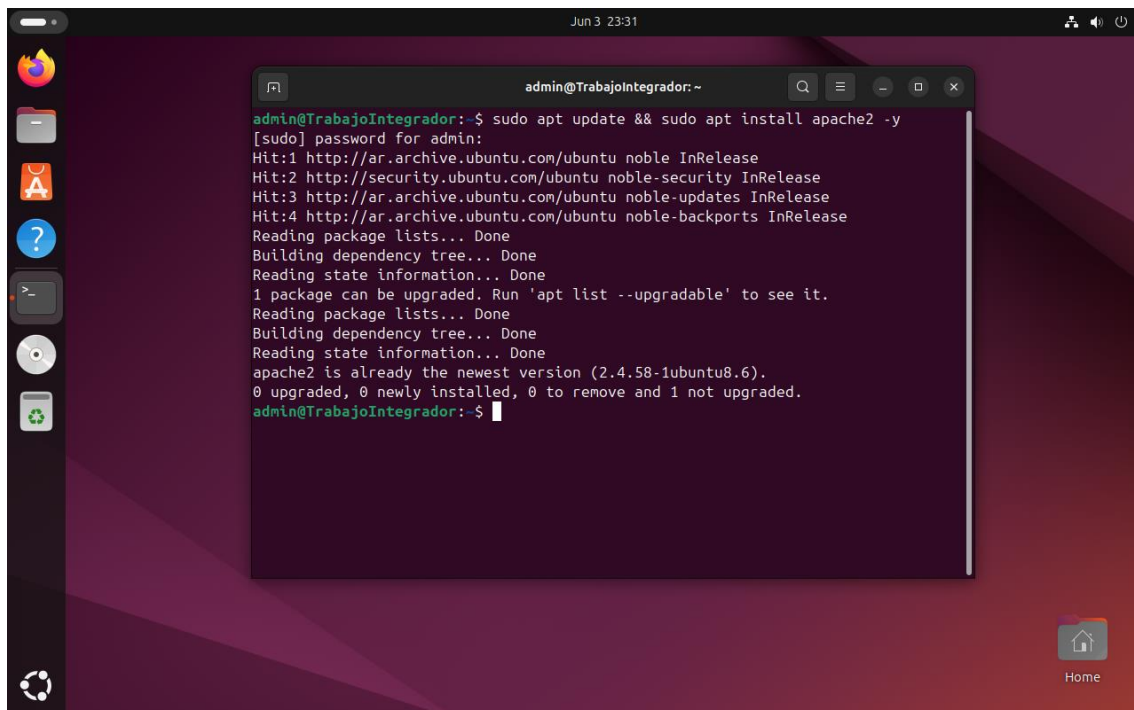
C. Guardar los cambios.



3.Instalacion de Apache en Ubuntu Server:

A. Ejecutar en la terminal.

sudo apt update && sudo apt install apache2 -y

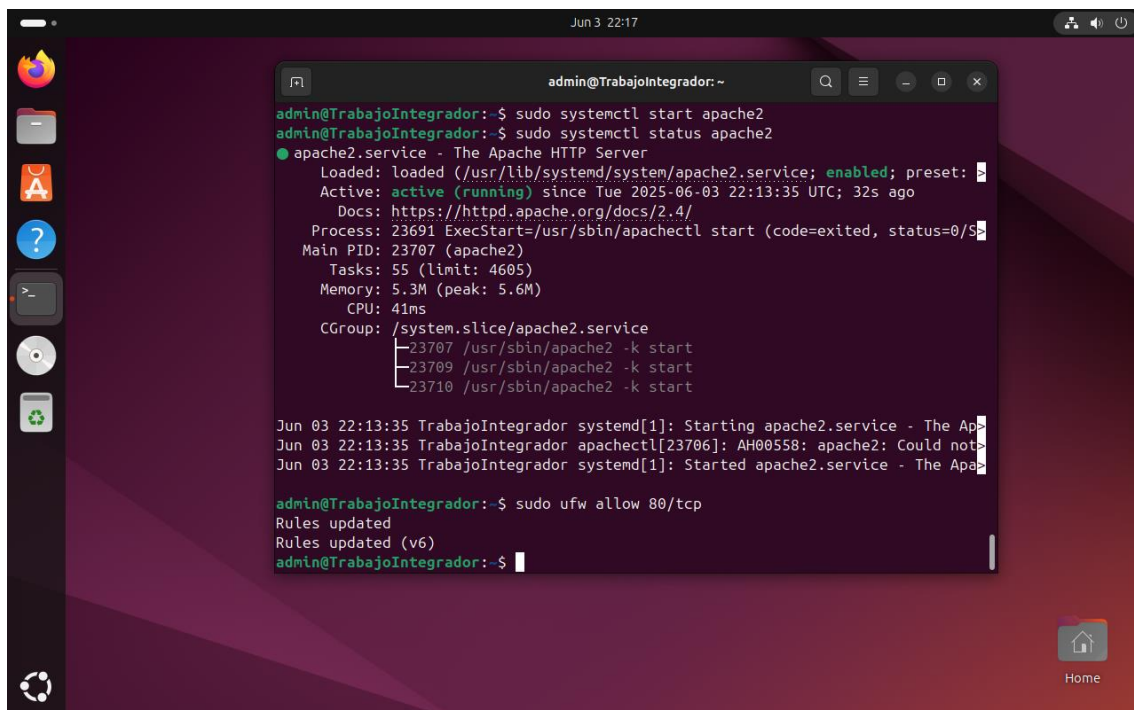


```
admin@TrabajoIntegrador:~$ sudo apt update && sudo apt install apache2 -y
[sudo] password for admin:
Hit:1 http://ar.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://ar.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://ar.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.6).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
admin@TrabajoIntegrador:~$
```

B. Verifica el estado del servicio y comprobar si está activo.

```
sudo systemctl status apache2
```

```
sudo systemctl start apache2
```



```
admin@TrabajoIntegrador:~$ sudo systemctl start apache2
admin@TrabajoIntegrador:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset:
   Active: active (running) since Tue 2025-06-03 22:13:35 UTC; 32s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 23691 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S
   Main PID: 23707 (apache2)
      Tasks: 55 (limit: 4605)
     Memory: 5.3M (peak: 5.6M)
        CPU: 41ms
    CGroup: /system.slice/apache2.service
            └─23707 /usr/sbin/apache2 -k start
              └─23709 /usr/sbin/apache2 -k start
                └─23710 /usr/sbin/apache2 -k start

Jun 03 22:13:35 TrabajoIntegrador systemd[1]: Starting apache2.service - The Ap
Jun 03 22:13:35 TrabajoIntegrador apachectl[23706]: AH00558: apache2: Could not
Jun 03 22:13:35 TrabajoIntegrador systemd[1]: Started apache2.service - The Apa

admin@TrabajoIntegrador:~$ sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)
admin@TrabajoIntegrador:~$
```

4. Configuración del Firewall UFW

A. Permitir tráfico web en los puertos 80 y 443:

```
sudo ufw allow 80/tcp
```

```
sudo ufw allow 443/tcp
```

```
sudo ufw enable
```

A terminal window on a Linux desktop. The terminal shows the output of starting the Apache2 service, followed by running 'sudo ufw allow 80/tcp', 'sudo ufw allow 443/tcp', and 'sudo ufw reload'. The firewall is not yet enabled.

```
admin@TrabajoIntegrador: ~  
Docs: https://httpd.apache.org/docs/2.4/  
Process: 23691 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S  
Main PID: 23707 (apache2)  
Tasks: 55 (limit: 4605)  
Memory: 5.3M (peak: 5.6M)  
CPU: 41ms  
CGroup: /system.slice/apache2.service  
├─23707 /usr/sbin/apache2 -k start  
├─23709 /usr/sbin/apache2 -k start  
└─23710 /usr/sbin/apache2 -k start  
  
Jun 03 22:13:35 TrabajoIntegrador systemd[1]: Starting apache2.service - The Ap  
Jun 03 22:13:35 TrabajoIntegrador apachectl[23706]: AH00558: apache2: Could not  
Jun 03 22:13:35 TrabajoIntegrador systemd[1]: Started apache2.service - The Apa  
  
admin@TrabajoIntegrador:~$ sudo ufw allow 80/tcp  
Rules updated  
Rules updated (v6)  
admin@TrabajoIntegrador:~$ sudo ufw allow 443/tcp  
Rules updated  
Rules updated (v6)  
admin@TrabajoIntegrador:~$ sudo ufw reload  
Firewall not enabled (skipping reload)  
admin@TrabajoIntegrador:~$
```

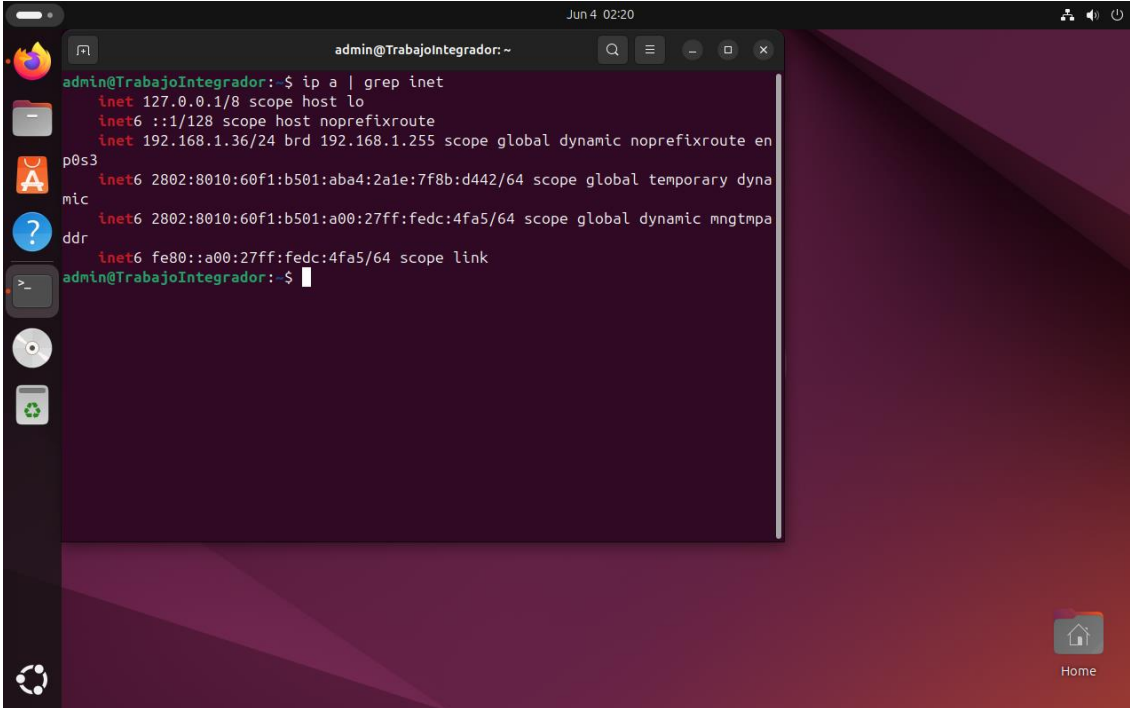
A terminal window on a Linux desktop. The terminal shows the continuation of the UFW configuration, including 'sudo ufw enable' which activates the firewall, and 'sudo ufw status' which shows the active rules for ports 80 and 443.

```
admin@TrabajoIntegrador: ~  
Jun 03 22:13:35 TrabajoIntegrador apachectl[23706]: AH00558: apache2: Could not  
Jun 03 22:13:35 TrabajoIntegrador systemd[1]: Started apache2.service - The Apa  
  
admin@TrabajoIntegrador:~$ sudo ufw allow 80/tcp  
Rules updated  
Rules updated (v6)  
admin@TrabajoIntegrador:~$ sudo ufw allow 443/tcp  
Rules updated  
Rules updated (v6)  
admin@TrabajoIntegrador:~$ sudo ufw reload  
Firewall not enabled (skipping reload)  
admin@TrabajoIntegrador:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
admin@TrabajoIntegrador:~$ sudo ufw status  
Status: active  
  
To Action From  
--  
80/tcp ALLOW Anywhere  
443/tcp ALLOW Anywhere  
80/tcp (v6) ALLOW Anywhere (v6)  
443/tcp (v6) ALLOW Anywhere (v6)  
  
admin@TrabajoIntegrador:~$
```

5. Verificación del IP del servidor.

A. Verificar IP asignada

`ip a | grep inet`



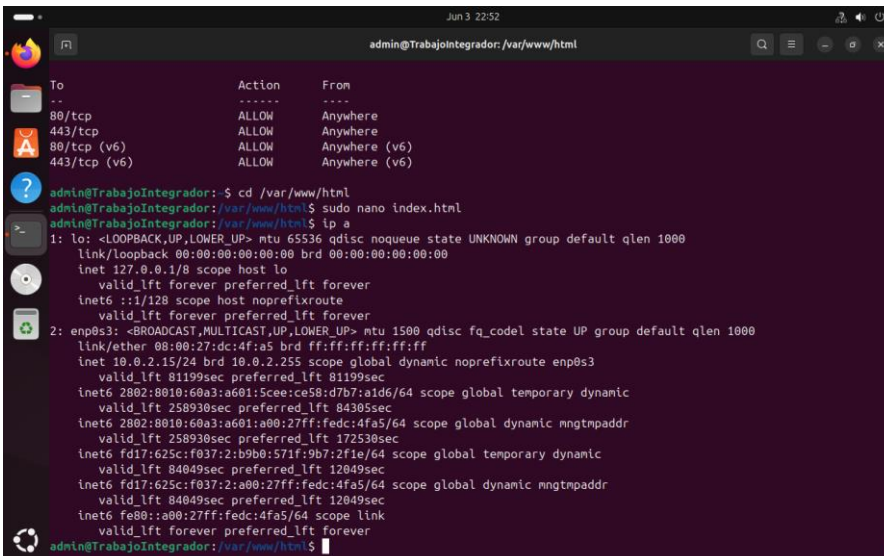
```
admin@TrabajoIntegrador:~$ ip a | grep inet
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host noprefixroute
inet 192.168.1.36/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
inet6 2802:8010:60f1:b501:aba4:2a1e:7f8b:d442/64 scope global temporary dynamic
inet6 2802:8010:60f1:b501:a00:27ff:fedc:4fa5/64 scope global dynamic mngtmpadr
inet6 fe80::a00:27ff:fedc:4fa5/64 scope link
```

6. Creación de una página web personalizada.

A. Editar el archivo index.html en /var/www/html/.

`sudo nano /var/www/html/`

`index.html`



```
admin@TrabajoIntegrador:/var/www/html$ sudo nano index.html
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:dc:4f:a5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 81199sec preferred_lft 81199sec
    inet6 2802:8010:60a3:a601:5cee:ce58:d7b7:a1d6/64 scope global temporary dynamic
        valid_lft 258930sec preferred_lft 84305sec
    inet6 2802:8010:60a3:a601:a00:27ff:fedc:4fa5/64 scope global dynamic mngtmpadr
        valid_lft 258930sec preferred_lft 172530sec
    inet6 fd17:625c:f037:2:b9b0:571f:9b7:2f1e/64 scope global temporary dynamic
        valid_lft 84049sec preferred_lft 12049sec
    inet6 fd17:625c:f037:2:a00:27ff:fedc:4fa5/64 scope global dynamic mngtmpadr
        valid_lft 84049sec preferred_lft 12049sec
    inet6 fe80::a00:27ff:fedc:4fa5/64 scope link
        valid_lft forever preferred_lft forever
```

B. Le agregamos nuevo contenido.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>Mi Servidor Apache</title>
```

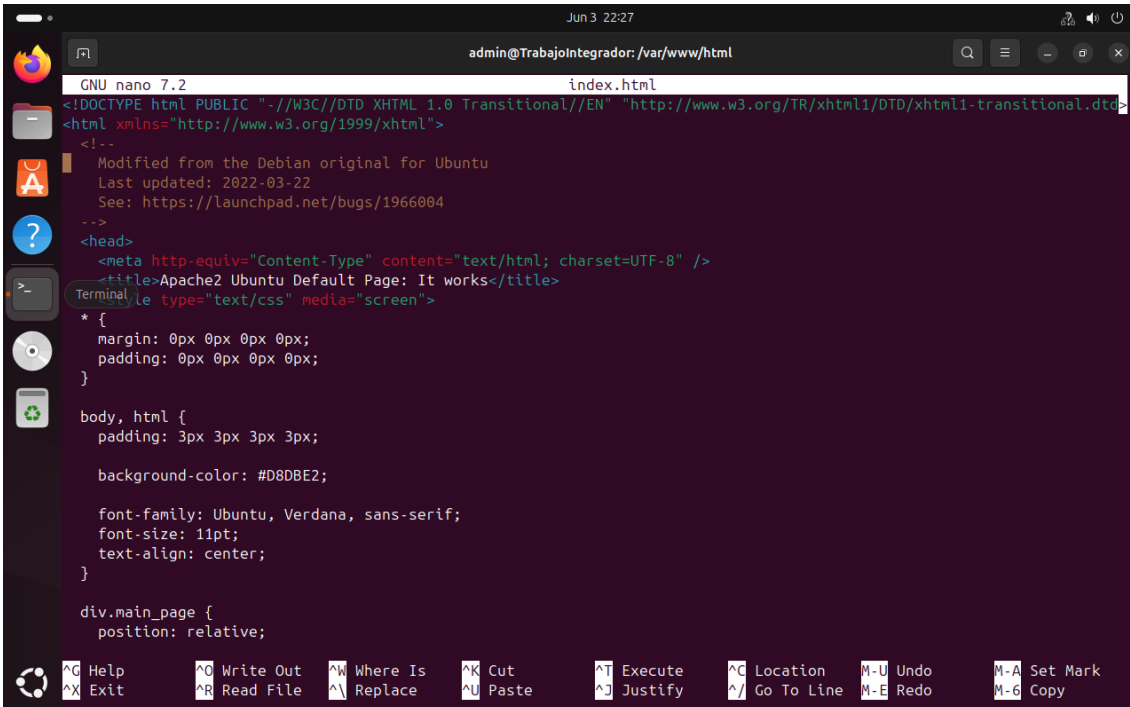
```
</head>
```

```
<body>
```

```
    <h1>¡Apache funcionando correctamente!</h1>
```

```
</body>
```

```
</html>
```



```
Jun 3 22:27
admin@Trabajointegrador: /var/www/html
GNU nano 7.2 index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
  Modified from the Debian original for Ubuntu
  Last updated: 2022-03-22
  See: https://launchpad.net/bugs/1966004
-->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Apache2 Ubuntu Default Page: It works</title>
<style type="text/css" media="screen">
* {
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}
body, html {
  padding: 3px 3px 3px 3px;
  background-color: #D8DBE2;
  font-family: Ubuntu, Verdana, sans-serif;
  font-size: 11pt;
  text-align: center;
}
div.main_page {
  position: relative;
}
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location ^U Undo      ^M Set Mark
^X Exit      ^R Read File ^\ Replace  ^U Paste     ^J Justify  ^/_ Go To Line ^E Redo     ^G Copy
```

```
GNU nano 7.2 index.html *
<!DOCTYPE html>
<html>

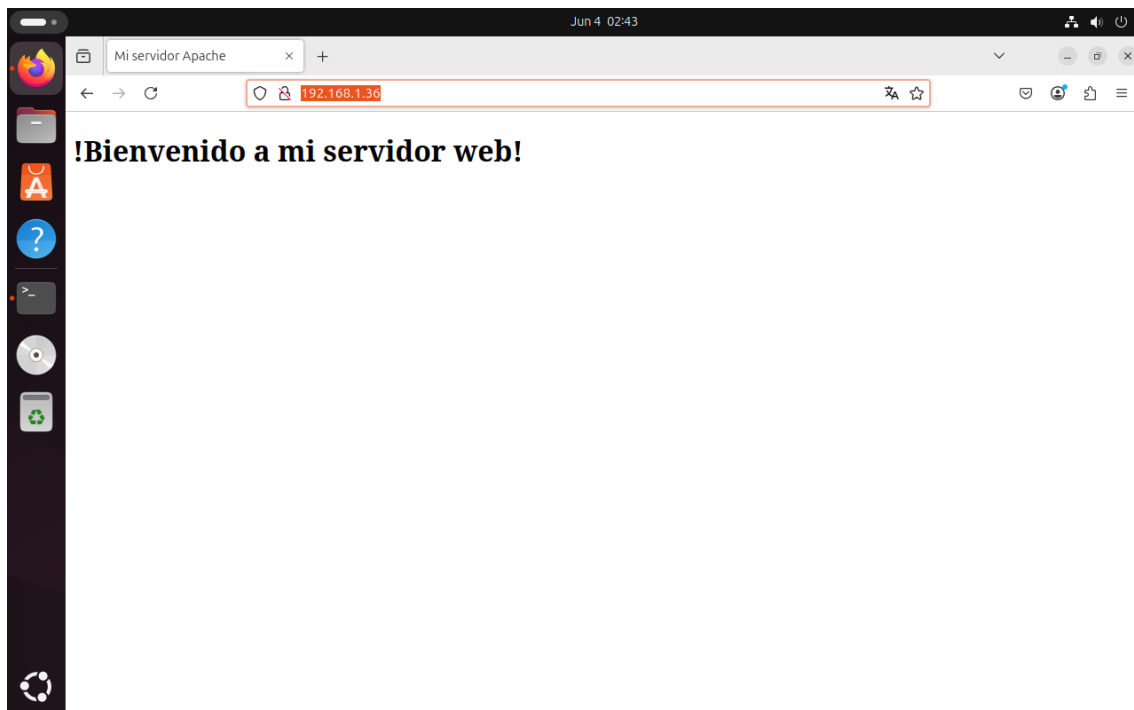
<head>

  <title>Mi servidor Apache</title>
</head>
<body>
  <h1>¡Bienvenido a mi servidor web!</h1>

</body>
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location   ^U Undo       ^A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify   ^_ Go To Line  ^-E Redo      ^-G Copy
```

C. Guardamos lo editado.

7. Verificación de Apache desde el Navegador.



4. Metodología Utilizada:

Para empezar con este trabajo, primero investigamos sobre el tema de la virtualización, ya que queríamos entender bien de qué se trataba y cómo aplicarlo. Buscamos información en sitios oficiales, documentación de Ubuntu, VirtualBox y páginas relacionadas con servidores web y redes.

Una vez que tuvimos la parte teórica clara, pasamos a la parte práctica. Lo que hicimos fue:

- Crear una máquina virtual con VirtualBox, usando una imagen ISO de Ubuntu Server 22.04. Le asignamos 4 GB de memoria RAM y 30 GB de disco.
- Configuramos la red en modo “Adaptador puente” para poder acceder a la máquina desde la red local.
Instalamos Apache desde la terminal, actualizando primero los paquetes y luego instalando el servidor con el comando `sudo apt install apache2 -y`.
- Configuramos el firewall (UFW) para permitir el tráfico por los puertos 80 y 443, que son los necesarios para el acceso web.
- Buscamos la dirección IP del servidor con el comando `ip a | grep inet`, y la usamos para probar el acceso desde un navegador.
- Editamos el archivo `index.html` en `/var/www/html/` usando nano, y ahí escribimos una página simple para verificar que Apache estuviera funcionando.

En cuanto al trabajo en grupo, nos repartimos las tareas: uno se encargó de hacer el caso práctico y el otro de la escritura del informe. Por último, desarrollamos el video explicativo.

5. Resultados Obtenidos:

- Servidor Apache funcionando de manera eficiente.
- Corrección de problemas de red y firewall.
- Visualización de página web desplegada correctamente.

6. Conclusiones:

A través del recorrido de este trabajo practico integrador aprendimos que la virtualización es una herramienta que nos ayuda a crear entornos de prueba, esenciales para nuestro aprendizaje sin tener la necesidad de utilizar hardware adicional. Las herramientas que se nos brindan son muy accesibles, confiables y potentes para realizar este propósito. Por ejemplo, la VirtualBox que nos permite crear una simulación de una maquina virtual con distintos sistemas operativos sin utilizar muchos recursos de hardware y poder realizar nuestro trabajo de manera efectiva.

7. Bibliografía:

- <https://www.virtualbox.org/>
- <https://ubuntu.com/download/desktop>
- <https://aws.amazon.com/es/what-is/virtualization/#:~:text=La%20virtualizaci%C3%B3n%20es%20una%20tecnol%C3%ADa,en%20una%20%20C3%BArica%20m%C3%A1quina%20f%C3%A1cil%20de%20usar>
- <https://httpd.apache.org/docs/2.4/>
- <https://help.ubuntu.com/community/UFW>

8. Anexos:

- Capturas de pantalla sobre la ejecución de la maquina virtual.