

Tarea 9: Administración de Redes en Ubuntu 22.04 LTS

Francisco Javier Sueza Rodríguez

21 de mayo de 2023

Centro:	IES Aguadulce
Ciclo Formativo:	Desarrollo Aplicaciones Web (Distancia)
Asignatura:	Sistemas Informáticos
Tema:	Tema 9 - Administración de Redes en Ubuntu 22.04 LTS

Índice

1	Caso Práctico	4
2	Ejercicios	4
2.1	Actividad 1: Configuración del Entorno de Red en Ubuntu	4
2.1.1	Enunciado	4
2.1.2	Solución	4
2.2	Actividad 2: Correo y Búsqueda de Información	7
2.2.1	Enunciado	7
2.2.2	Solución	8
2.3	Actividad 3: Compartir recursos en la red	9
2.3.1	Enunciado	9
2.3.2	Solución	9
2.4	Actividad 4: Conexión Remota a Dispositivos	11
2.4.1	Enunciado	11
2.4.2	Solución	12
2.5	Actividad 5: Servicio FTP	15
2.5.1	Enunciado	15
2.5.2	Solución	15
2.6	Actividad 6: Servidor Web	18
2.6.1	Enunciado	18
2.6.2	Solución	19

Índice de figuras

2.1	Configuración ip estática con Netplan	5
2.2	Configuración de red en la máquina virtual	6
2.3	Configuración de red en el SO hosts	6
2.4	Ping desde la MV a la puerta de enlace	6
2.5	Ping desde máquina anfitriona a la MV	7
2.6	Ping desde la MV a la máquina anfitriona	7
2.7	Artículo documentación oficial Ubuntu	8
2.8	Correo desde la plataforma al profesor	8
2.9	Creación de la carpeta compartida	9
2.10	Carpeta creada y compartiéndose en el anfitrión	10
2.11	Creación de un fichero de texto en la carpeta compartida	11
2.12	Lectura del fichero desde la máquina anfitrión	11
2.13	Ventana de conexión del cliente SSH	12
2.14	Conexión realiza con ssh a la MV	13
2.15	Configuración del servidor de VNC en Gnome	13
2.16	Configuración de la conexión con KRDC	14
2.17	Conexión realizada desde la máquina anfitriona con KRDC	14
2.18	Instalación de vsftpd en la máquina virtual	15
2.19	Configuración de vsftpd	16
2.20	Conexión al FTP con Filezilla desde la máquina anfitrión	17
2.21	Descarga y Subida de archivos con Filezilla	18
2.22	Página web de ejemplo	18
2.23	Instalación de Apache en la máquina virtual	19
2.24	Habilitación de Apache para que se ejecute al inicio	20
2.25	Copia de index.html y cambio de permisos	20
2.26	Acceso a la página web desde el anfitrión	21

1. Caso Práctico

La empresa recibe el encargo de configurar servicios de red. Para ello, Ada asigna el trabajo a Antonio y Juan. En principio, se establece que serán cuatro los servicios de red a configurar. Además, se deberá configurar el entorno de red de manera adecuada.

2. Ejercicios

2.1. Actividad 1: Configuración del Entorno de Red en Ubuntu

2.1.1. Enunciado

Configura la MV de Ubuntu para que tenga siempre la misma dirección IP. Es decir, vamos a configurar de forma estática su dirección IP. Se puede hacer mediante comandos o con la interfaz gráfica (GUI) de Ubuntu. Para realizar esta actividad se recomienda consultar el siguiente documento: [Configuración de red en Ubuntu](#) (pdf).

La dirección IP asignada tiene que pertenecer a la red a través de la cual se comunicarán la máquina anfitriona y la virtual. Comprobarás que todo está correcto con un ping desde la MV a la puerta de enlace por defecto de tu red, un ping desde la máquina anfitriona a la virtual y otro desde la virtual a la anfitriona.

Las **capturas** deben mostrar, al menos:

- Configuración de red de la máquina virtual.
- Configuración de red del equipo anfitrión.
- Ping positivo desde la MV a la puerta de enlace por defecto de tu red desde una terminal de Ubuntu.
- Ping positivo desde la máquina anfitriona a la MV.
- Ping positivo desde la MV a la anfitriona.

2.1.2. Solución

En este primer ejercicio vamos a asignar una **IP statica** a la interfaz ethernet de la Máquina Virtual. Ya que el sistema es una Ubuntu 22.04, debemos usar **Netplan**, la herramienta que usa actualmente Ubuntu y que sustituye a las net-tools tradicionales. Personalmente hubiera preferido usar el método tradicional, ya que es más genérico y puede usarse en cualquier distribución, pero como estamos estudiando Ubuntu, usaremos las herramientas que ésta incluye.

Netplan es una abstracción para un conjunto de aplicaciones que nos ayudan a configurar y obtener información de nuestra red y nuestros dispositivos de red, usa ficheros de configuración con formato **YAML** y funciona conjuntamente con **systemd-network**.

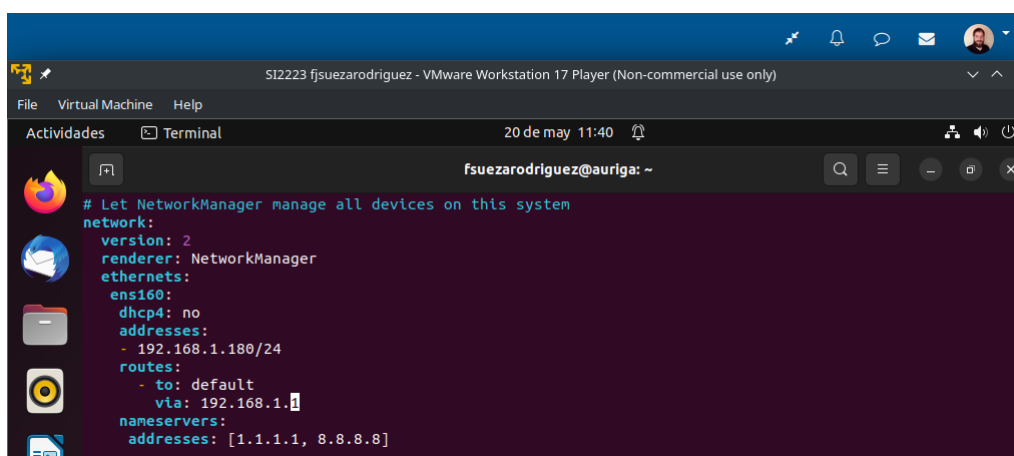
Para establecer la dirección ip estática de la interfaz de red, debemos editar el fichero **/etc/netplan/01-network-manager-all.yaml**, que es donde reside la configuración de red administrada por esta herramienta. Antes de eso podemos recopilar la información que necesitaremos con el comando **ip**, así sabremos la dirección actual de la interfaz de red, la subred a la que pertenece, dirección de broadcast, etc..

Una vez que sabemos esa información, usando nuestro editor preferido (nano, emacs, vi..) abrimos el fichero mencionado, teniendo en cuenta que debemos tener privilegios de root, por lo que incluiremos el comando **sudo** delante del editor que vayamos a usar, en nuestro caso el comando quedaría así:

```
sudo vi /etc/netplan/01-network-manager-all.yaml
```

Si queremos, como precaución, podemos hacer una **copia de seguridad del fichero** de configuración, simplemente usando el comando **cp**, por si tocamos algo que no debemos que podamos restaurarlo sin problema.

Una vez abierto el fichero, vamos a introducir los **datos de nuestra configuración de red**, los cuales podemos ver en la siguiente captura con más detalle.

A screenshot of a terminal window within a VMware Workstation 17 Player. The terminal shows the contents of the file /etc/netplan/01-network-manager-all.yaml. The configuration is for a network interface named ens160, set to use static IP addressing with the address 192.168.1.180/24. It also shows a default route and DNS servers at 1.1.1.1 and 8.8.8.8. The terminal prompt is fsuezarodriguez@auriga: ~.

```
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernet:
    ens160:
      dhcp4: no
      addresses:
        - 192.168.1.180/24
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses: [1.1.1.1, 8.8.8.8]
```

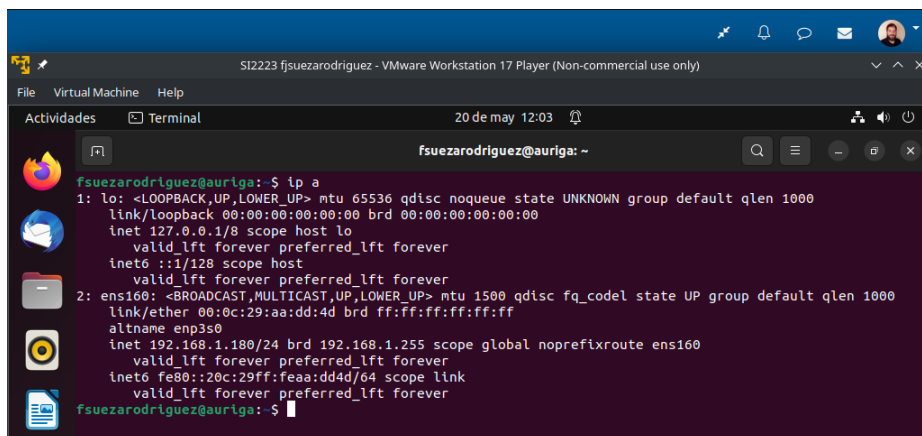
Figura 2.1: Configuración ip estática con Netplan

Solo un par de **comentarios** sobre la configuración que hemos puesto. La opción **gateway4** para indicar la puerta de enlace esta ya obsoleta, por lo que hemos tenido que usar **routes** con la opción **default** para indicar la puerta de enlace hacia internet. Además, hemos aprovechado para cambiar el **servidores DNS** y que nuestra interfaz use los proporcionados por **Cloudflare**, los cuales suelen ser bastante rápidos.

Tras guardar el fichero de configuración, hemos usado el comando **netplan try** para ver si la configuración es correcta, como ha sido nuestro caso.

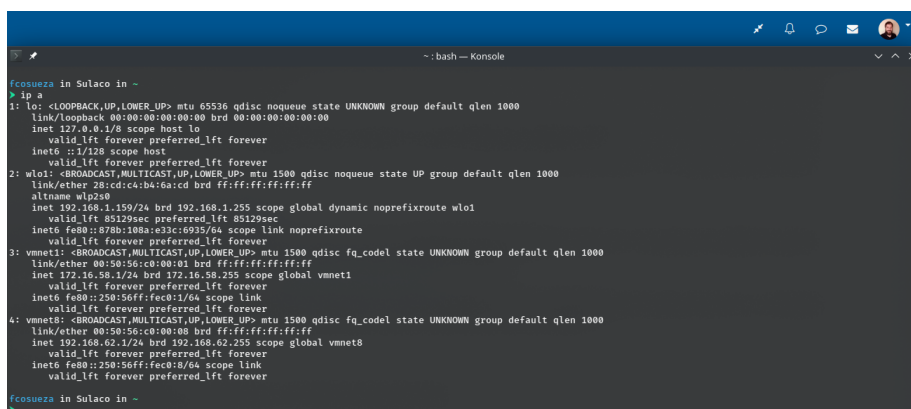
Para comprobar que la configuración de red se ha **realizado correctamente**, hemos llevado a cabo los siguientes pasos:

1. En primer lugar hemos **reiniciado** la máquina virtual, para comprobar que al reiniciarla la configuración de red es persistente, y se mantiene la dirección IP estática que le habíamos asignado.
2. A continuación, hemos usado el comando **ip a** tanto en la **máquina virtual** como en el **SO operativo host**, que también es una Ubuntu 22.04 LTS. El resultado de ambos comandos los podemos ver en la siguientes capturas de pantalla.



```
fsuezarodriguez@auriga:~$ ip a
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
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:aa:dd:4d brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.1.180/24 brd 192.168.1.255 scope global noprefixroute ens160
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feaa:dd4d/64 scope link
        valid_lft forever preferred_lft forever
fsuezarodriguez@auriga:~$
```

Figura 2.2: Configuración de red en la máquina virtual



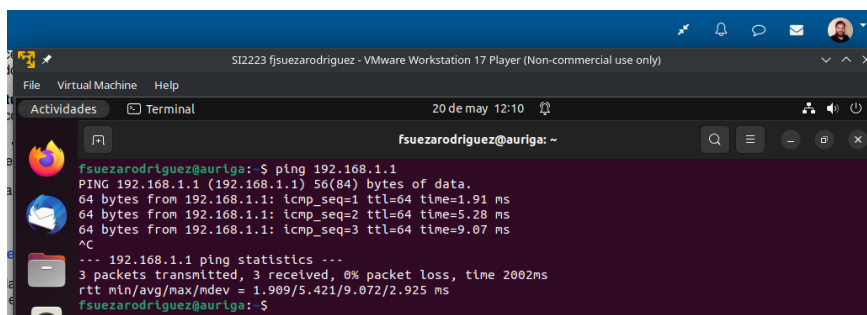
```
fcsueza in Sulaco in ~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 28:c0:c4:b4:6a:cd brd ff:ff:ff:ff:ff:ff
    altname wlp2s0
    inet 192.168.1.159/24 brd 192.168.1.255 scope global dynamic noprefixroute wlo1
        valid_lft 85129sec preferred_lft 85129sec
    inet6 fe80::8780:108a:e3c:6935/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: vmmnet1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 00:50:56:c0:00:00 brd ff:ff:ff:ff:ff:ff
    inet 172.16.58.1/24 brd 172.16.58.255 scope global vmmnet1
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fec0:0/64 scope link
        valid_lft forever preferred_lft forever
4: vmmnet8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 00:50:56:c0:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.02.1/24 brd 192.168.02.255 scope global vmmnet8
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fec0:0/64 scope link
        valid_lft forever preferred_lft forever
fcsueza in Sulaco in ~$
```

Figura 2.3: Configuración de red en el SO hosts

Como vemos en la figura 2.2, la configuración de red de nuestra máquina virtual está correctamente.

3. El siguiente paso será hacer **ping** a diferentes destinos desde la máquina virtual para comprobar que la conexión funciona correctamente. Los ping que hemos realizado son los siguientes:

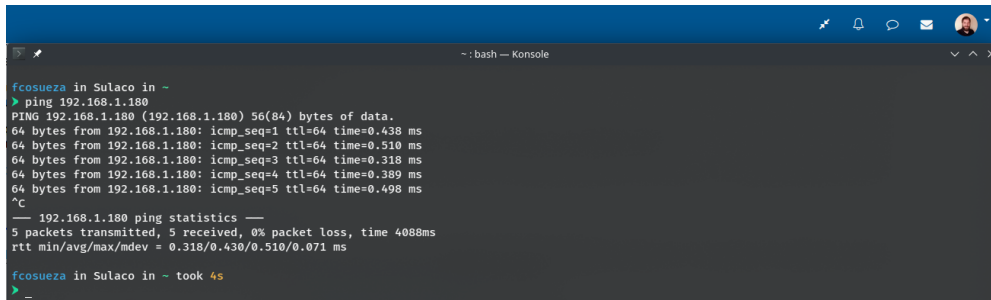
- Desde la MV a la puerta de enlace:



```
fsuezarodriguez@auriga:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data:
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.91 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=5.28 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=9.07 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.909/5.421/9.072/2.925 ms
fsuezarodriguez@auriga:~$
```

Figura 2.4: Ping desde la MV a la puerta de enlace

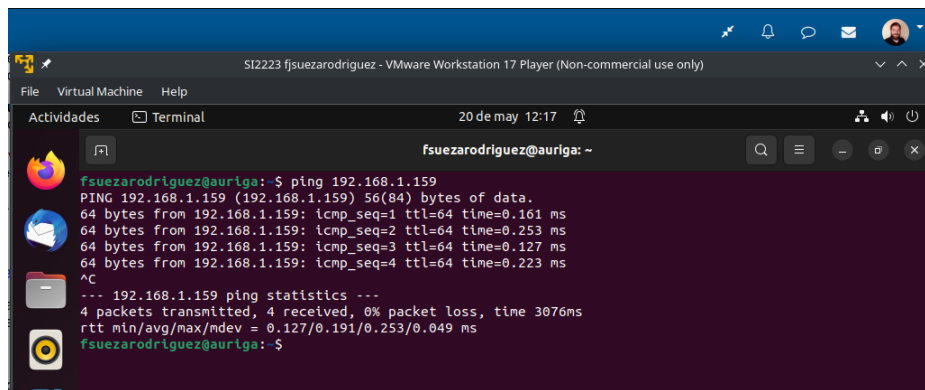
■ Desde la máquina anfitriona a la MV:



```
fcosueza in Sulaco in ~  
> ping 192.168.1.180  
PING 192.168.1.180 (192.168.1.180) 56(84) bytes of data.  
64 bytes from 192.168.1.180: icmp_seq=1 ttl=64 time=0.438 ms  
64 bytes from 192.168.1.180: icmp_seq=2 ttl=64 time=0.510 ms  
64 bytes from 192.168.1.180: icmp_seq=3 ttl=64 time=0.318 ms  
64 bytes from 192.168.1.180: icmp_seq=4 ttl=64 time=0.389 ms  
64 bytes from 192.168.1.180: icmp_seq=5 ttl=64 time=0.498 ms  
^C  
--- 192.168.1.180 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4088ms  
rtt min/avg/max/mdev = 0.318/0.430/0.510/0.071 ms  
fcosueza in Sulaco in ~ took 4s
```

Figura 2.5: Ping desde máquina anfitriona a la MV

■ Desde la MV a la máquina anfitriona:



```
fsuezarodriguez@auriga: ~  
$ ping 192.168.1.159  
PING 192.168.1.159 (192.168.1.159) 56(84) bytes of data.  
64 bytes from 192.168.1.159: icmp_seq=1 ttl=64 time=0.161 ms  
64 bytes from 192.168.1.159: icmp_seq=2 ttl=64 time=0.253 ms  
64 bytes from 192.168.1.159: icmp_seq=3 ttl=64 time=0.127 ms  
64 bytes from 192.168.1.159: icmp_seq=4 ttl=64 time=0.223 ms  
^C  
--- 192.168.1.159 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3076ms  
rtt min/avg/max/mdev = 0.127/0.191/0.253/0.049 ms  
fsuezarodriguez@auriga: ~$
```

Figura 2.6: Ping desde la MV a la máquina anfitriona

Como podemos comprobar, todas las conexiones funcionan correctamente en ambos sentidos. Aunque no se incluye aquí la captura, también se ha realizado ping a servidores externos a la red local desde la MV para comprobar que hay conexión a internet correctamente, siendo los resultados positivos.

2.2. Actividad 2: Correo y Búsqueda de Información

2.2.1. Enunciado

Desde la MV, haz login en el aula virtual de Sistemas Informáticos y envía a tu profesor un correo a través de la plataforma (icono del “sobre” a la izquierda de tu foto de perfil de usuario). El correo tendrá como asunto “<Tu_Nombre>Tarea 9.- Documentación de Ubuntu 22.04”, y en el cuerpo del asunto habrá un breve texto a tu elección, y se incluirá un enlace a la página oficial de documentación de Ubuntu Desktop 22.04.

Las **capturas** deben mostrar, al menos:

- La página oficial de documentación de Ubuntu Desktop 22.04.
- El correo que se envía a través de la plataforma desde la MV redactado.

2.2.2. Solución

En primer lugar, hemos buscado un artículo en la documentación oficial de Ubuntu, el concreto el artículo **About Gnome**, el cual podemos ver en la siguiente captura.

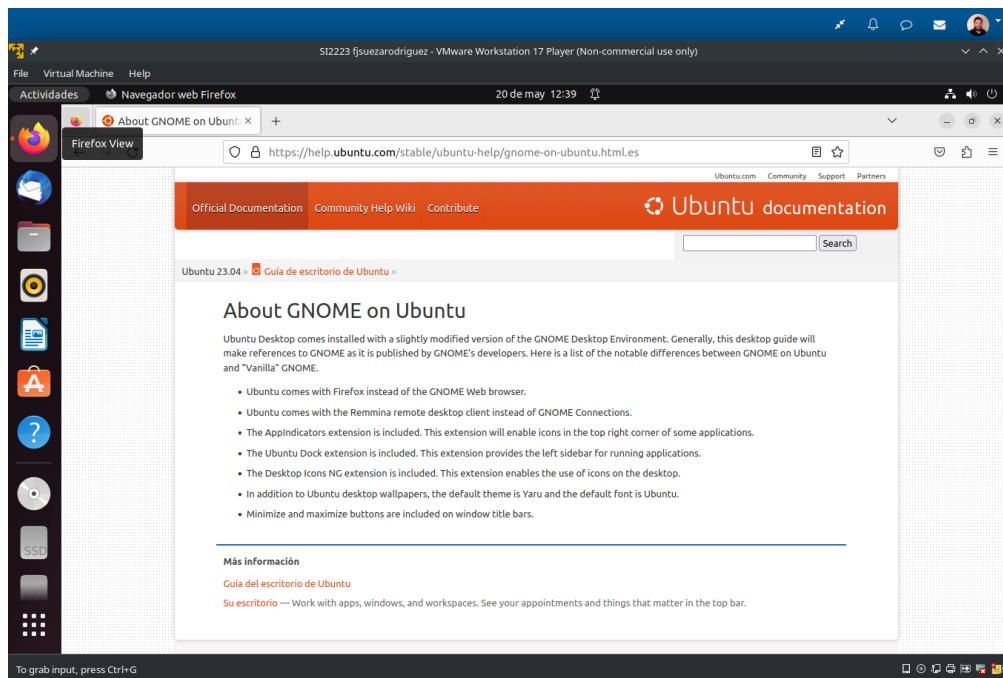


Figura 2.7: Artículo documentación oficial Ubuntu

A continuación, hemos **conectado a la plataforma** y **redactado el mensaje** al profesor con los datos que se especifican en el enunciado, como vemos a continuación.

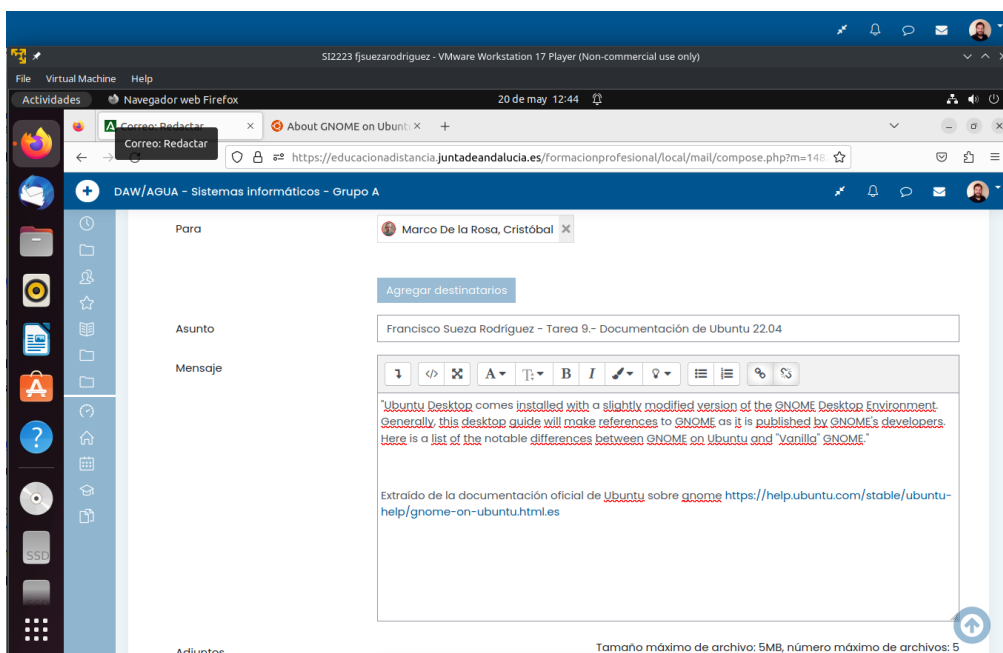


Figura 2.8: Correo desde la plataforma al profesor

2.3. Actividad 3: Compartir recursos en la red

2.3.1. Enunciado

Crea en tu máquina anfitriona una carpeta y compártela en red. A continuación, accede a dicha carpeta desde la máquina virtual Ubuntu y crea un fichero de texto nuevo en su interior con tu nombre. **Se debe hacer a través de la red local** y en ningún caso usando la opción de carpetas compartidas que ofrezca el software de virtualización usado.

Esta operación suele ser muy simple y no debería requerir ninguna instalación, pero dependiendo de qué SO tengas en tu máquina anfitriona es posible que necesites instalar en ella o en la MV distintos paquetes (Samba, NFS, APFS, etc.).

Esquema de conexión: Máquina virtual con Ubuntu >Carpeta compartida en SO anfitrión.

Las **capturas** deben mostrar, al menos:

- Creación y compartición de la carpeta en la máquina real.
- Acceso a ella desde la MV. Se debe ver y explicar claramente cómo se ha hecho el acceso.
- Creación y edición de un fichero de texto desde la MV.

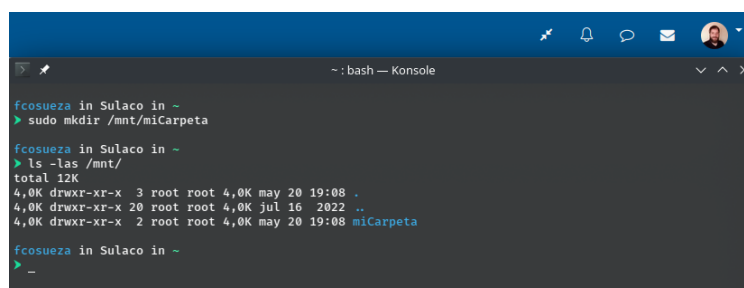
2.3.2. Solución

En este ejercicio vamos a **compartir una carpeta** entre la máquina virtual y el sistema host.

Ya que las **dos máquinas** tienen sistemas operativos **Linux**, ambas con Linuxs, vamos a usar el servidor **NFS**. Podríamos también usar Samba, pero como NFS es específico para compartir carpetas entre sistemas Linux nos hemos decantado por este.

Para **compartir la carpeta** entre los dos sistemas se han llevado a cabo los siguientes pasos:

1. En primer lugar, hemos instalado el paquete **nfs-kernel-server** en la máquina anfitriona, ya que es donde vamos a crear la carpeta que se va a compartir. Esto se he realizado usando el comando **apt-get**.
2. En segundo lugar, hemos creado la carpeta que vamos a compartir en el sistema anfitrión. En concreto, la carpeta creada ha sido **/mnt/miCarpeta**. Para ello sea ha usado el comando **mkdir**, como podemos ver en la siguiente captura.



```
fcosueza in Sulaco in ~  
> sudo mkdir /mnt/miCarpeta  
fcosueza in Sulaco in ~  
> ls -las /mnt/  
total 12K  
4,0K drwxr-xr-x  3 root root 4,0K may 20 19:08 .  
4,0K drwxr-xr-x 20 root root 4,0K jul 16  2022 ..  
4,0K drwxr-xr-x  2 root root 4,0K may 20 19:08 miCarpeta  
fcosueza in Sulaco in ~  
> _
```

Figura 2.9: Creación de la carpeta compartida

3. A continuación hemos editado el archivo **/etc/exports**, para permitir el acceso a la carpeta compartida que hemos creado desde la máquina virtual. Para ello, hemos añadido la siguiente línea:

```
/mnt/miCarpeta 192.168.1.180/24(rw,async,no_subtree_check,root_squash)
```

4. Como **último paso** en el **sistema anfitrión**, hemos montado el sistema **exportfs** y **reiniciado el servicio nfs**, usando el comando **systemctl restart nfs-kernel-server**. Tras esto, nuestra carpeta estará creada correctamente y compartiéndose en la red local, como podemos ver en la siguiente captura.

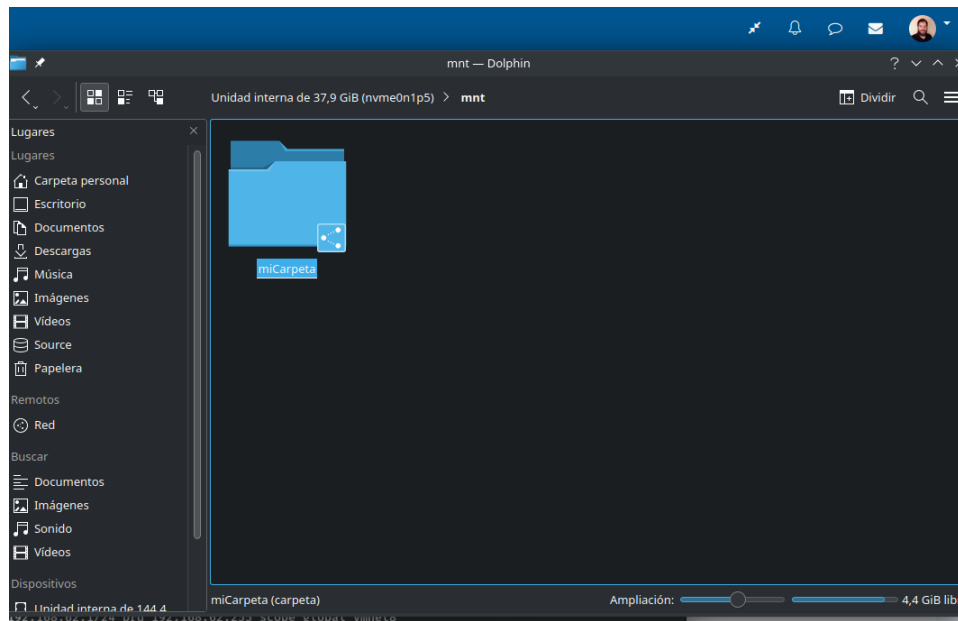


Figura 2.10: Carpeta creada y compartiéndose en el anfitrión

5. Para poder **acceder desde el cliente** (nuestra máquina virtual), en primer lugar debemos instalar el paquete **nfs-common**, lo cual hemos llevado a cabo usando el comando **apt-get**.
6. A continuación hemos creado la carpeta **/mnt/miCarpeta-local**, que será el punto donde montaremos la carpeta compartida, para poder acceder a ella de forma local como si estuviera en nuestro propio ordenador.

Este tipo de acceso es más sencillo de realizar en NFS que en Samba, que tiene más restricciones, uno de los motivos por los cuales hemos elegido NFS.
7. El siguiente paso ha sido **montar la carpeta remota** en el directorio que hemos creado, usando para ello el comando **mount -t nfs 192.168.1.159:/mnt/miCarpeta /mnt/miCarpeta-local**. Con esto, ya tenemos montada la carpeta remota en nuestro sistema con permisos de escritura y de lectura.
8. Para comprobar que todo funciona correctamente, hemos creado un fichero de texto simple usando el comando **echo** y redireccionando la salida a un fichero de texto desde nuestra máquina virtual, como podemos ver en la siguiente captura.

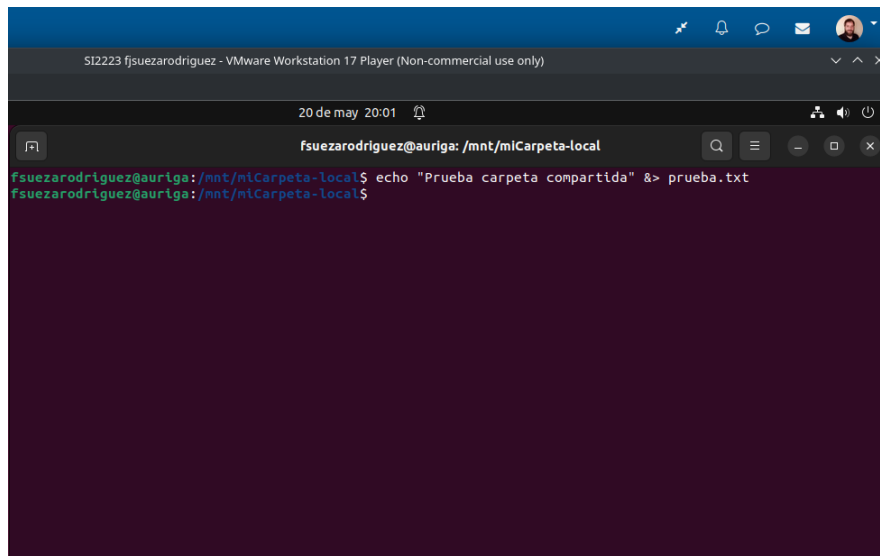


Figura 2.11: Creación de un fichero de texto en la carpeta compartida

9. Por último, hemos leído el fichero de texto desde la máquina anfitrión para comprobar que se ha creado correctamente. Hemos usado el comando **cat** desde la terminal, como vemos a continuación.

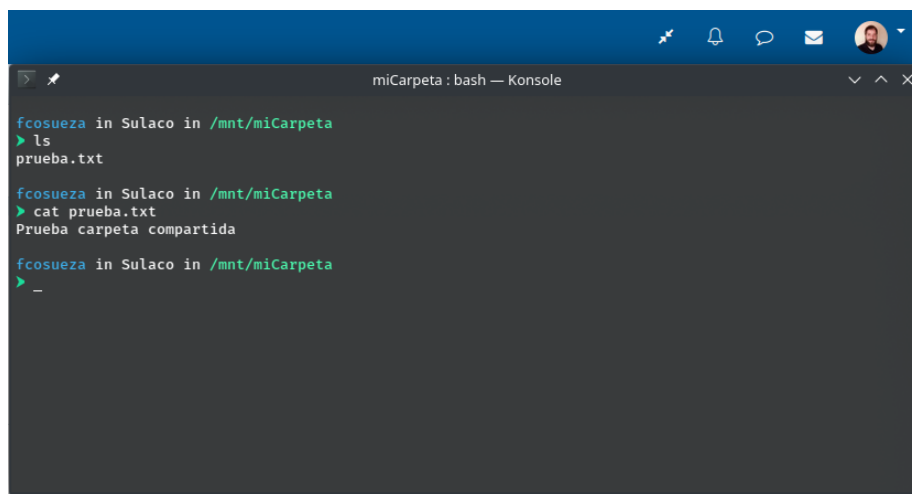


Figura 2.12: Lectura del fichero desde la máquina anfitrión

2.4. Actividad 4: Conexión Remota a Dispositivos

2.4.1. Enunciado

Realiza conexiones remotas desde el SO anfitrión (cliente) a la MV de Ubuntu (servidor) usando dos mecanismos:

- SSH. Comenta las ventajas de utilizar SSH frente a Telnet.
- VNC.

Esquema de conexión 1: Cliente SSH en la máquina anfitriona >Servidor SSH en MV Ubuntu. Esquema de conexión 2: Cliente de VNC en la máquina anfitriona >Servidor VNC en MV Ubuntu.

Las **capturas** deben mostrar, al menos:

- Ventana de conexión del cliente SSH en la máquina anfitriona.
- Sesión SSH ya conectada y funcionando.
- Configuración de compartición de pantalla en la MV para el acceso mediante VNC.
- Ventana de conexión del cliente VNC en la máquina anfitriona.
- Sesión VNC ya conectada y funcionando.

2.4.2. Solución

En este ejercicio vamos a realizar una conexión remota desde la máquina anfitriona a la máquina virtual de dos formas diferentes. En primer lugar, usaremos **ssh** (Secure Shell) y a continuación usaremos VCN (Virtual Network Computing).

- **Conexión con SSH**

Para realizar este tipo de conexión entre las dos máquinas nos necesitamos realizar ninguna configuración especial en ninguna de ellas. La aplicación **OpenSSH** viene **instalada por defecto** en todas las distribuciones Linux y además está configurado por defecto para los casos básicos. Lo único que deberíamos hacer es inicializar el servicio en la máquina virtual empleando el comando **systemctl**.

Una vez inicializado el servicio en la MV, desde la máquina anfitriona solo deberíamos realizar la conexión, utilizando el comando **ssh fsuezarodriguez@192.168.1.180**, como podemos ver en la siguiente captura de pantalla.

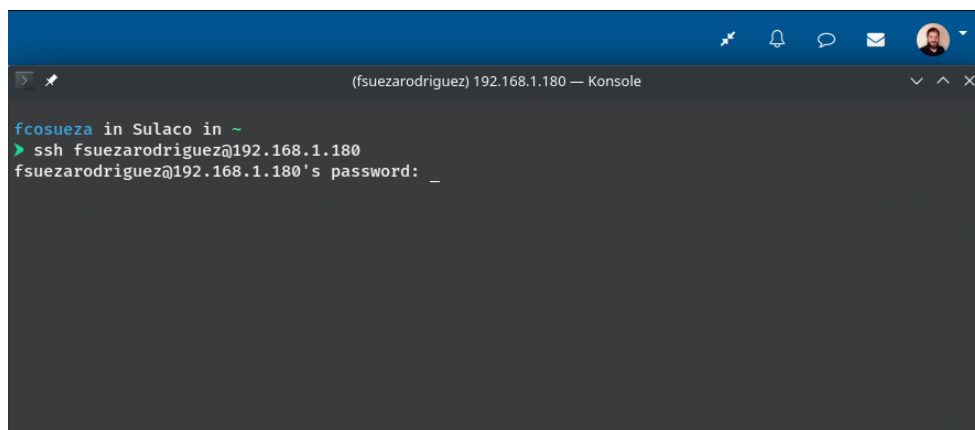
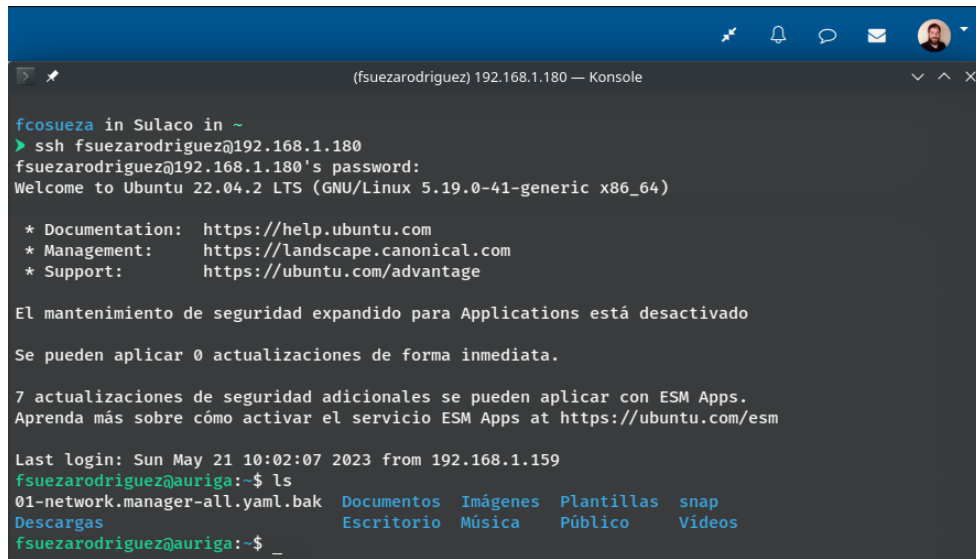


Figura 2.13: Ventana de conexión del cliente SSH

Para realizar la conexión, nos pedirá la **contraseña del usuario** que hemos indicado. Una vez introducida la contraseña, se realizará la conexión situándonos en el directorio home del usuario.

En la siguiente captura podemos ver como se ha establecido la conexión y hemos realizado un listado de los archivos en el directorio del usuario fsuezarodriguez. Podemos ver además como el

prompt ha cambiado y nos muestra que estamos trabajando con el **usuario fsuezarodriguez** en el **host auriga**.



```
(fsuezarodriguez) 192.168.1.180 — Konsole

fcosueza in Sulaco in ~
> ssh fsuezarodriguez@192.168.1.180
fsuezarodriguez@192.168.1.180's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-41-generic x86_64)

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

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 0 actualizaciones de forma inmediata.

7 actualizaciones de seguridad adicionales se pueden aplicar con ESM Apps.
Aprenda más sobre cómo activar el servicio ESM Apps at https://ubuntu.com/esm

Last login: Sun May 21 10:02:07 2023 from 192.168.1.159
fsuezarodriguez@auriga:~$ ls
01-network.manager-all.yaml.bak  Documentos  Imágenes  Plantillas  snap
Descargas                        Escritorio  Música    Público     Vídeos
fsuezarodriguez@auriga:~$ _
```

Figura 2.14: Conexión realiza con ssh a la MV

■ Conexión con VNC

Ahora vamos a realizar una conexión utilizando el protocolo **VNC**, que nos permite conectarnos de forma remota al escritorio de la máquina servidor y manejarlo. Al igual que en el punto anterior, no vamos a tener que realizar instalación de ninguna aplicación, ya que las que vamos a usar, **vin**o (servidor) y **remmina** ya vienen instaladas por defecto.

En primer lugar, hemos realizado la **configuración** del acceso remoto mediante VNC en Ubuntu, que se realiza accediendo a **Configuración**→**Compartir**, como podemos ver en la siguiente captura.

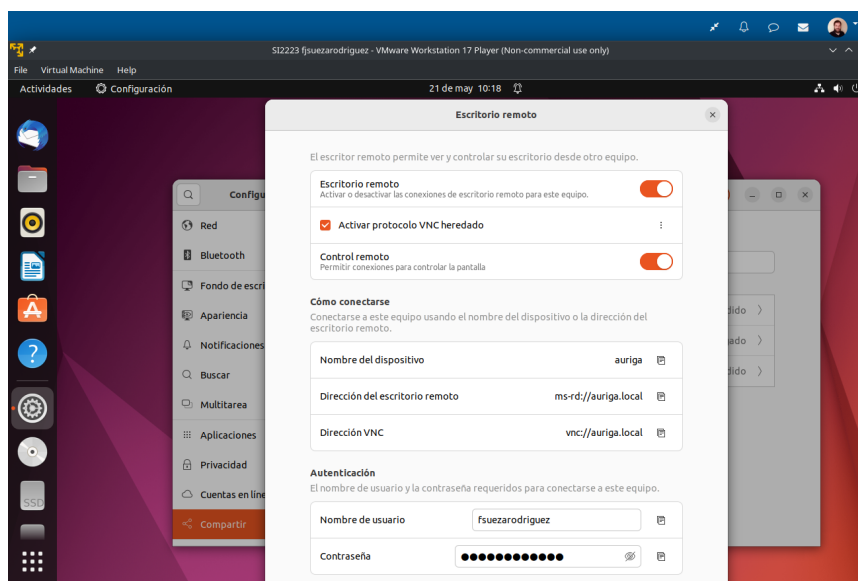


Figura 2.15: Configuración del servidor de VNC en Gnome

En la **máquina anfitrión** vamos a usar **KRDC**, ya la distribución que se usa en esta máquina es **Kubuntu 22.04 LTS** y es la aplicación que viene por defecto instalada.

En la siguiente captura podemos ver la **configuración de la conexión desde KRDC**, como vemos nos ofrece diferentes opciones, incluía usar ssh tunneling para realizar la conexión cifrada.

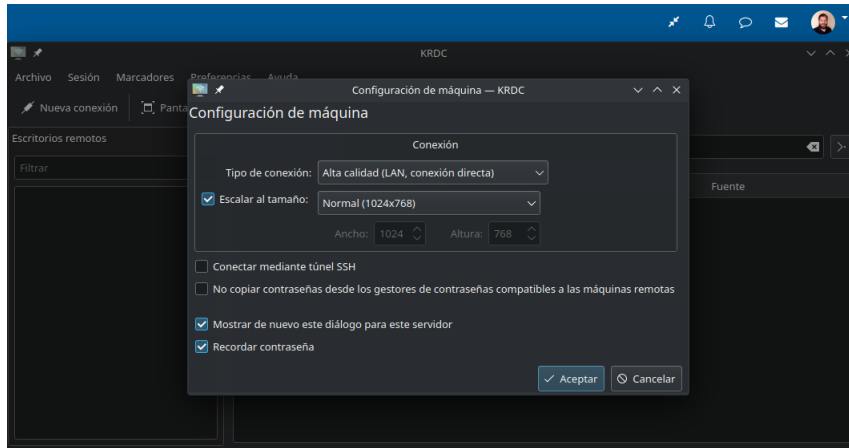


Figura 2.16: Configuración de la conexión con KRDC

Una vez introducidos los datos de configuración que queramos, pulsamos en aceptar y tras pedirnos la contraseña, que deberemos introducir la que hayamos establecido en la máquina virtual, la **conexión se realizará con éxito**, si hemos hecho todo bien, como vemos en la siguiente imagen.

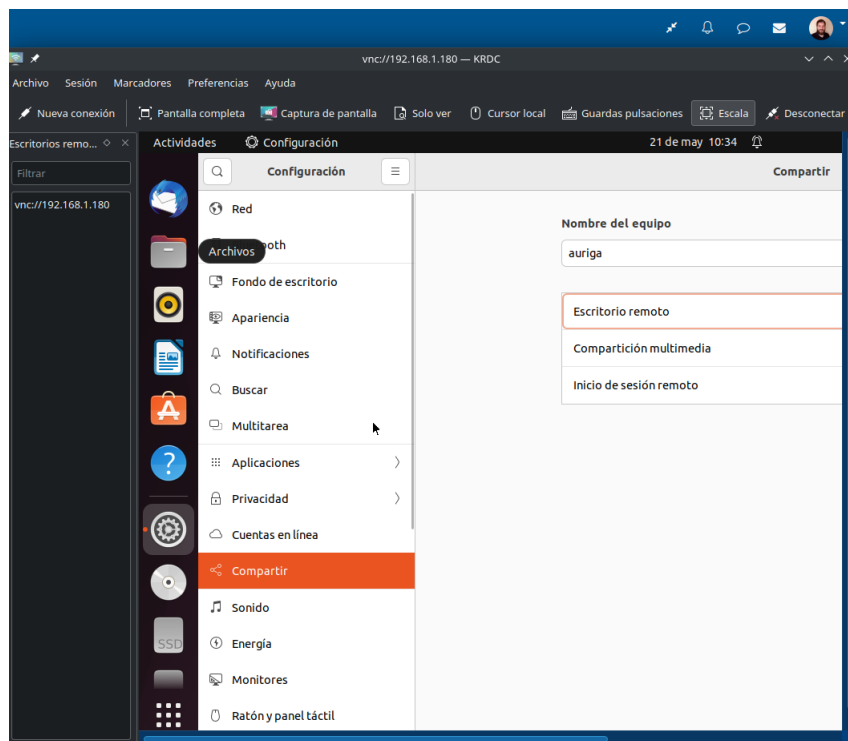


Figura 2.17: Conexión realizada desde la máquina anfitriona con KRDC

2.5. Actividad 5: Servicio FTP

2.5.1. Enunciado

Instala en la MV de Ubuntu un servidor FTP y en el SO anfitrión un cliente FTP como Filezilla, para acceder desde él al servidor. Una vez esté todo instalado, realiza una transmisión de algún archivo desde el cliente FTP al servidor y viceversa. Muestra los mensajes del cliente FTP en los que se confirma que la descarga y subida se han realizado correctamente.

Esquema de conexión: Cliente FTP en SO anfitrión > Servidor FTP en MV Ubuntu.

Las **capturas** deben mostrar, al menos:

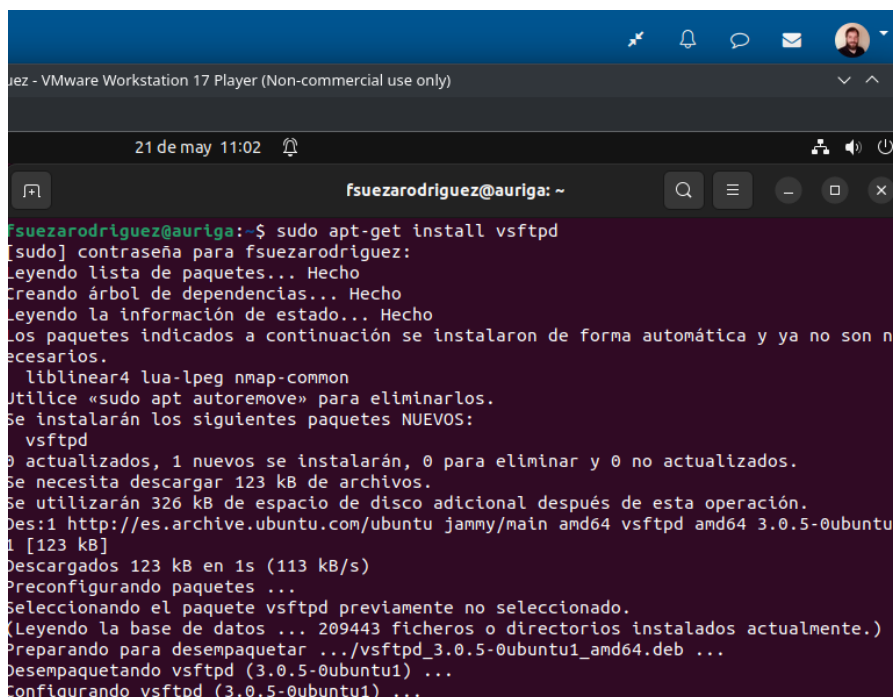
- Instalación servidor FTP en MV Ubuntu.
- Configuración del servidor FTP con los permisos necesarios para permitir acceso de lectura y escritura.
- Conexión desde el cliente FTP en la máquina anfitriona.
- Una descarga y una subida correctas.

2.5.2. Solución

Ahora vamos a montar un **servidor FTP** en la máquina virtual y vamos a acceder a él desde la máquina anfitriona. En el lado del servidor, vamos a instalar **vsftpd**, un servidor FTP ligero, robusto y fácil de configurar. En el lado del cliente vamos a usar **Filezilla**, uno de los clientes más usados.

Para llevar a cabo todo este proceso vamos a realizar los siguiente pasos:

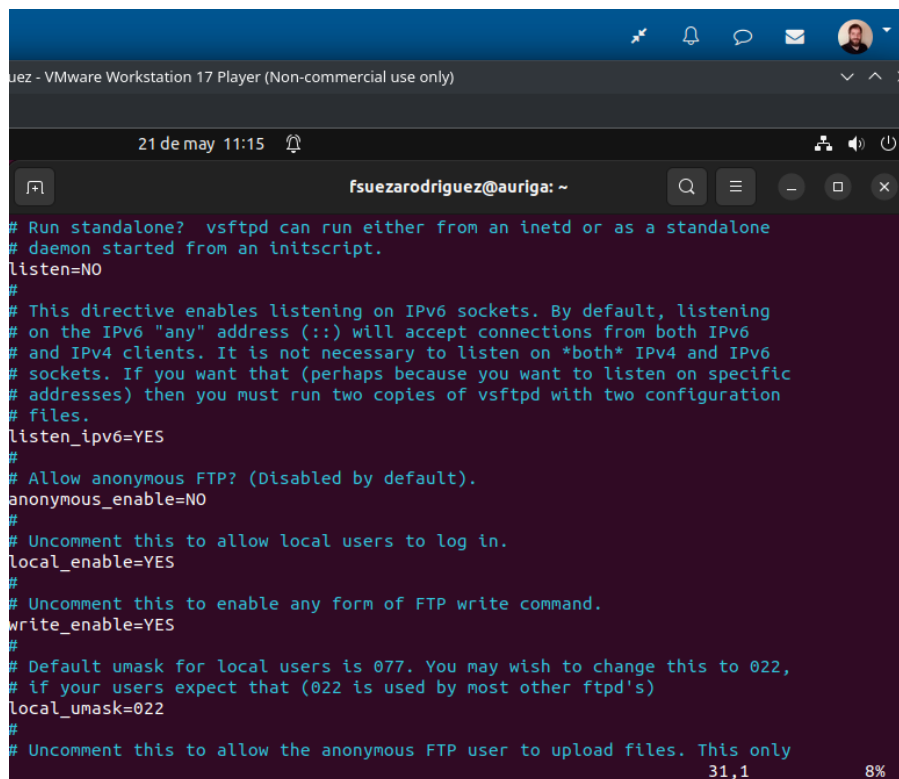
1. En primer lugar, vamos a **instalar vsftpd** en la máquina virtual, para lo que vamos a usar **apt-get**, como vemos en la siguiente captura.



```
fsuezarodriguez@auriga: ~  
fsuezarodriguez@auriga:~$ sudo apt-get install vsftpd  
[sudo] contraseña para fsuezarodriguez:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
liblinear4 lua-lpeg nmap-common  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes NUEVOS:  
vsftpd  
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
Se necesita descargar 123 kB de archivos.  
Se utilizarán 326 kB de espacio de disco adicional después de esta operación.  
Des:1 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 vsftpd amd64 3.0.5-0ubuntu1 [123 kB]  
Descargados 123 kB en 1s (113 kB/s)  
Preconfigurando paquetes ...  
seleccionando el paquete vsftpd previamente no seleccionado.  
(Leyendo la base de datos ... 209443 ficheros o directorios instalados actualmente.)  
Preparando para desempaquetar .../vsftpd_3.0.5-0ubuntu1_amd64.deb ...  
Desempaquetando vsftpd (3.0.5-0ubuntu1) ...  
Configurando vsftpd (3.0.5-0ubuntu1) ...
```

Figura 2.18: Instalación de vsftpd en la máquina virtual

2. El siguiente paso es **configurar el servidor FTP**, lo que vamos a hacer editando el **archivo de configuración** de vsftpd, que editaremos con el editor de textos **vim**, como vemos en la siguiente captura.



```
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
```

Figura 2.19: Configuración de vsftpd

A la hora de configurar el servidor, hemos decidido usar como **usuario** el que ya teníamos creado en la máquina virtual, esto es, **fsuezarodriguez**, por lo que al conectar desde la máquina anfitrión apareceremos en el directorio **/home/fsuezarodríguez**.

Podríamos haber creado otro usuario, llamado, por ejemplo, ftp, pero con el usuario que tenemos ya nos vale para lo que queremos hacer. Además, así nos evitamos complicar el proceso ya que habría que crear el usuario, grupo, modificar ciertos archivos, etc...

Las principales opciones que hemos usado en la configuración son las siguientes:

- **listen=NO**: no queremos que el servidor se ejecute al inicio, sino ejecutarlo nosotros manualmente. Es una cuestión de preferencias, pero si el servidor va a estar continuamente activo esta opción debería tener el valor **YES**.
- **listen_ipv6=YES**: con esta opción habilitamos la escucha en los sockets ipv6 y ipv4, para que acepte conexión tanto de una versión del protocolo como del otro.
- **anonymous_enable=NO**: con esta opción deshabilitamos el acceso anónimo al servidor.
- **write_enable=YES**: aquí estamos concediendo permisos de escritura al usuario del ftp. En el caso de que solo ofreciéramos archivos para descarga, esta opción tendría que tener el valor **NO**.
- **local_umask=022**: con esta directiva estamos estableciendo que los archivos creados por lo

usuarios tengan los **permisos 755**. Es una de las configuraciones mas usuales en los FTPs y con ello nos aseguramos de que cuando un usuario crea un archivo, no pueda ser borrado por otros usuarios, aunque si puede ser ejecutado y leído por estos.

Aunque hay más opciones, estas podríamos decir que son las más importantes.

3. El siguiente paso es **realizar una conexión** desde la máquina anfitriona al FTP. Para ello, como hemos comentado anteriormente, vamos a usar **Filezilla**.

Una vez introducidos los datos de conexión, usuario, servidor, contraseña, etc..., pulsamos en **conexión rápida** y la conexión se realizará correctamente, como podemos ver en la siguiente captura.

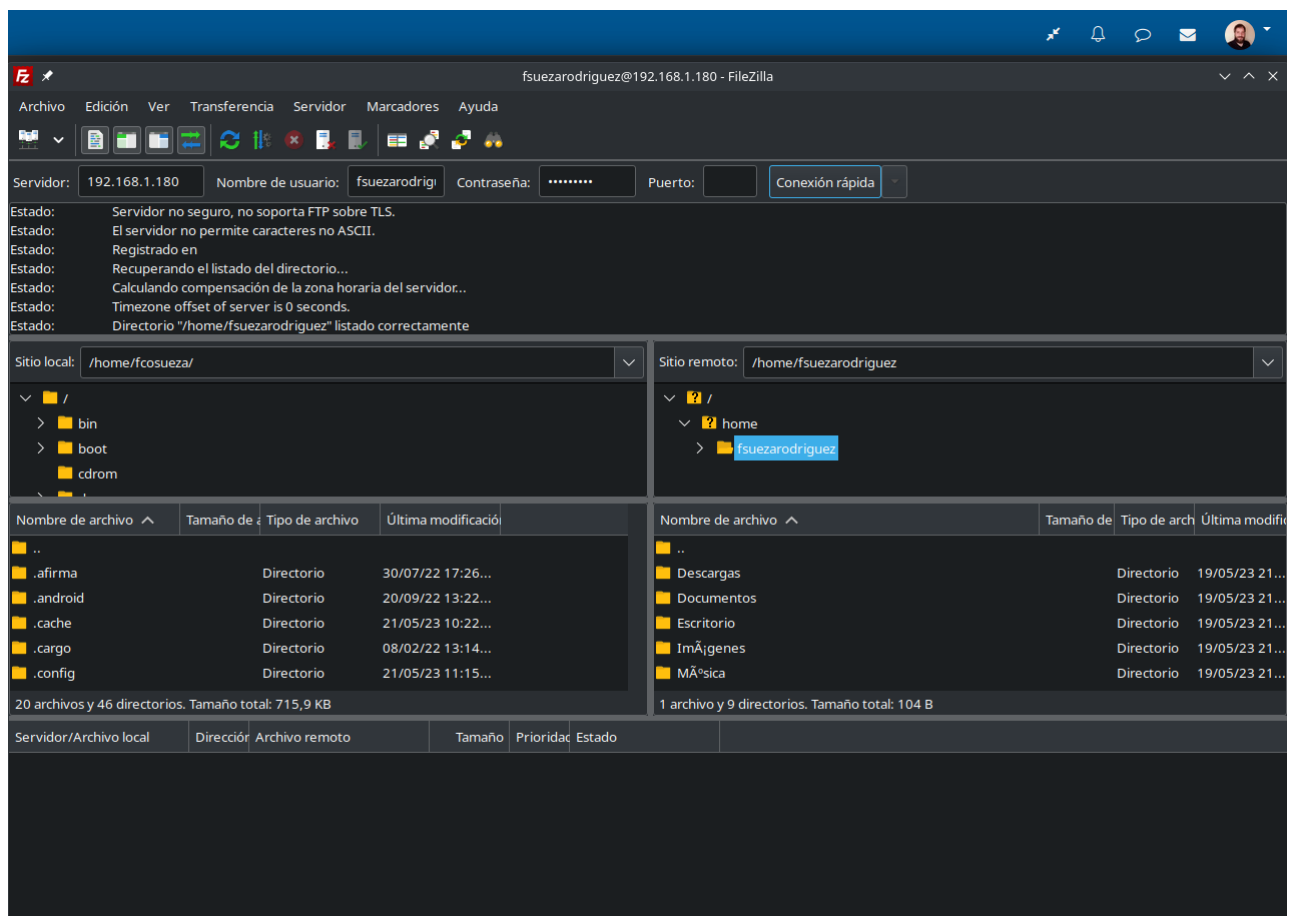


Figura 2.20: Conexión al FTP con Filezilla desde la máquina anfitrión

4. Como **último paso**, vamos a crear un fichero de texto, llamado **archivo.txt**, en el directorio del FTP desde la máquina virtual, para realizar su descarga desde Filezilla.

Al mismo tiempo, hemos creado otro archivo de texto en la **máquina anfitriona**, llamado **subida.txt**, que subiremos al servidor FTP usando también Filezilla.

En la siguiente captura, podemos ver como las dos operaciones (subida y bajada), se han realizado con éxito.

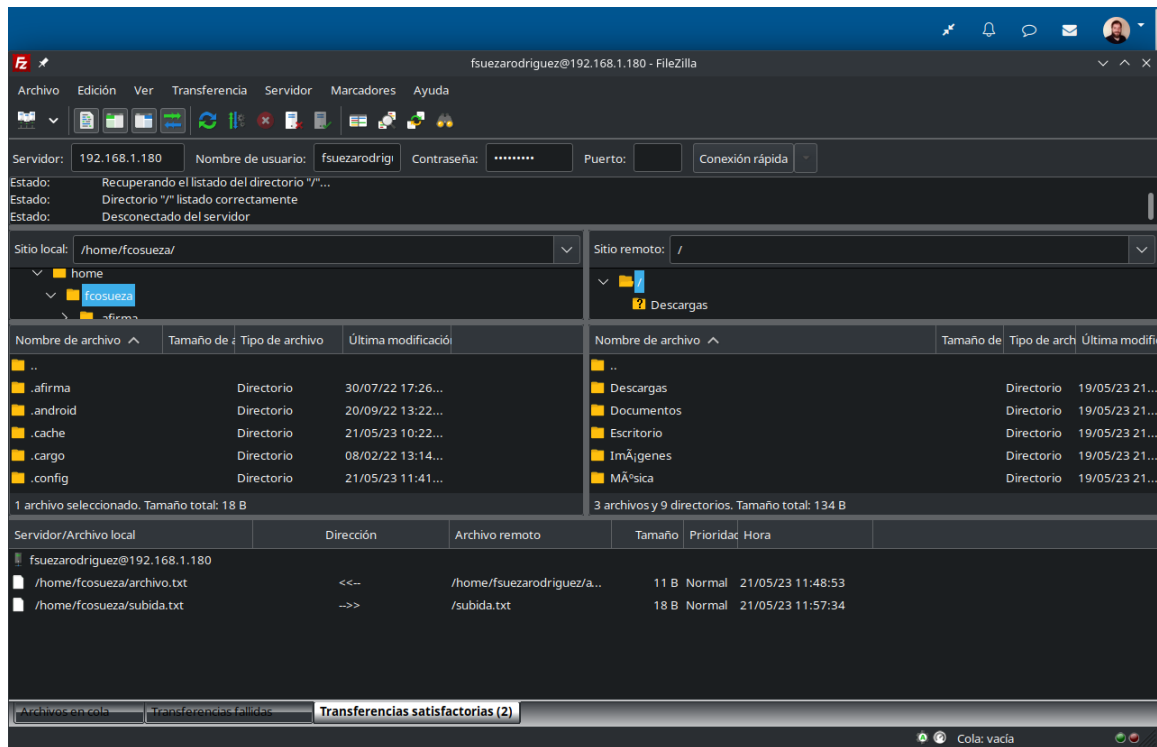


Figura 2.21: Descarga y Subida de archivos con Filezilla

2.6. Actividad 6: Servidor Web

2.6.1. Enunciado

Instala el servidor web Apache en la MV de Ubuntu y configúralo para que se inicie automáticamente al iniciar Ubuntu. A continuación, en el SO anfitrión abre en un navegador la página web miprimera-web.html que creaste en la tarea 6. Para ello previamente tendrás que copiar dicha página web en el directorio correspondiente del servidor Apache que has instalado.

Si no realizaste la actividad donde se creaba esta página web, a continuación se detalla el código de la misma que deberás crear y guardar en el archivo miprimera-web.html. Para ello utiliza un editor de texto de Ubuntu y en la imagen no olvides poner tu foto.

```

1 <html lang="es">
2 <head>
3 <title>CFG5 DAW/DAM - Módulo SI - Unidad 9</title>
4 <meta charset="UTF-8">
5 </head>
6 <body>
7 <h1>Tarea 9 de Sistemas Informáticos</h1>
8 <h2>Esta es mi primera página de prueba en código HTML</h2>
9 
10 Realizado por - Tu Nombre y Apellidos<br>
11 Creado el día - dd/mm/aaaa
12 <h2>Curso 2022/2023</h2>
13 </body>
14 </html>

```

Figura 2.22: Página web de ejemplo

Esquema de conexión: Navegador web en la máquina anfitriona > Servidor Apache en MV Ubuntu.

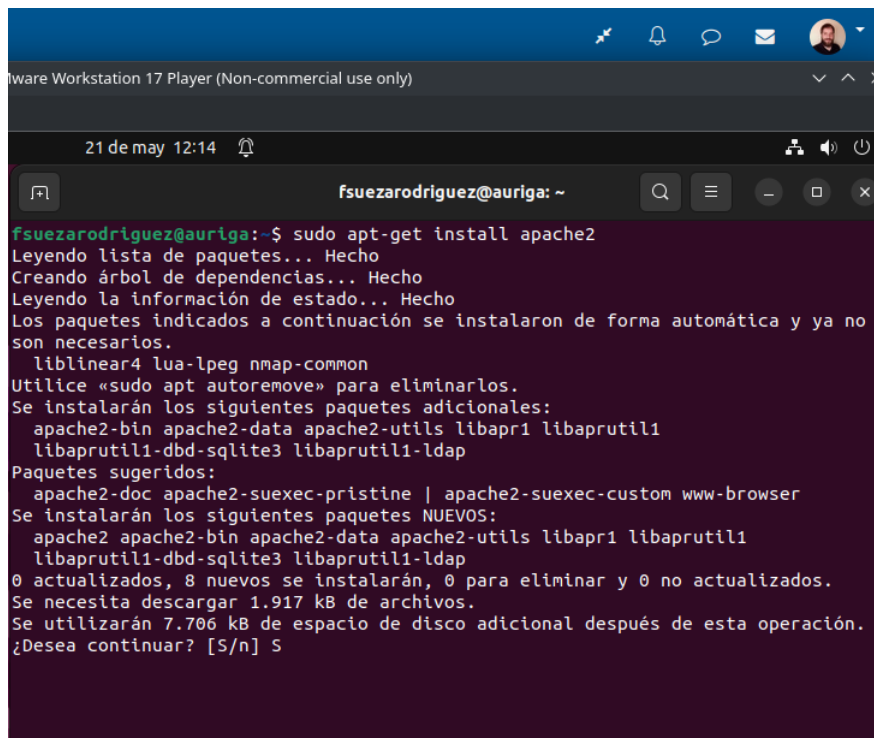
Las **capturas** deben mostrar, al menos:

- Instalación de Apache en la MV.
- Habilitar el servicio Apache para que se inicie automáticamente con el inicio de Ubuntu.
- Archivos de la página web copiados al directorio correspondiente de Apache (comentar/corregir permisos si es necesario).
- Acceso a la página web desde el navegador web en la máquina anfitriona (comentar/corregir permisos si es necesario).

2.6.2. Solución

Por último, vamos a **instalar y desplegar un servidor web** en la máquina virtual, usando el servidor **Apache**, uno de los servidores web más utilizados. La página web que vamos a servir es **HTML** plano, por lo que no será necesario instalar PHP. Los pasos que hemos seguido son los siguientes:

1. En primer lugar, hemos **instalado Apache2** en la máquina virtual, usando el comando **apt-get**, como vemos en la siguiente figura.

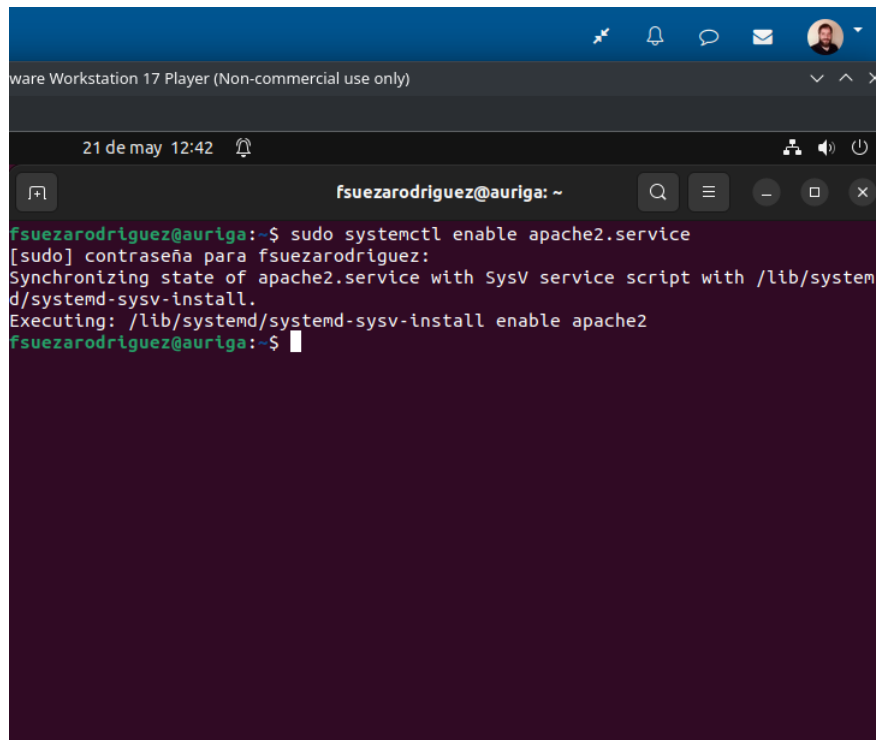


```
fsuezarodriguez@auriga: ~$ sudo apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  liblinear4 lua-lpeg nmap-common
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 1.917 kB de archivos.
Se utilizarán 7.706 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Figura 2.23: Instalación de Apache en la máquina virtual

2. Una vez que se ha instalado correctamente, vamos a indicar que se ejecute el servidor cada vez que se inicie el sistema operativo. Podríamos usar **sysv-rc-conf**, pero Apache tiene muy buena integración con **systemd**, por lo que su manejo es mucho más rápido usando los comandos **service** ó **systemctl**. Nosotros, vamos a usar **systemctl** para iniciar Apache cada vez que se inicie la máquina.

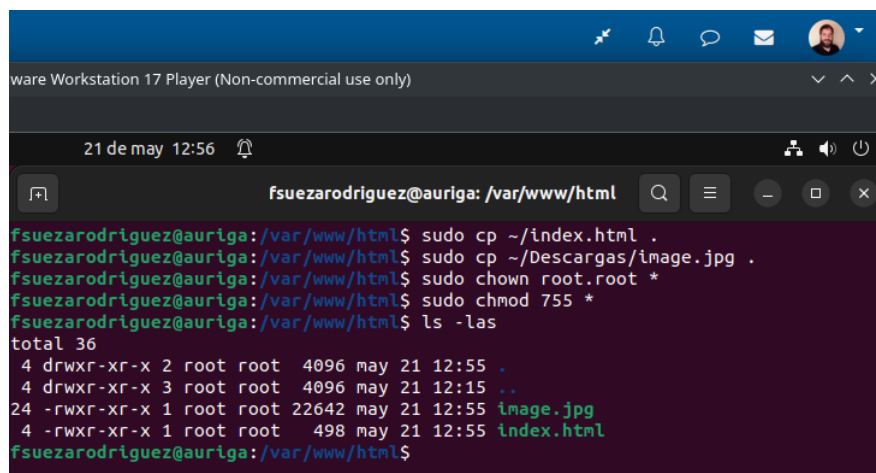
Para ello, vamos a usar el comando **systemctl enable apache2.service**, como podemos ver en la siguiente captura.



```
fsuezarodriguez@auriga:~$ sudo systemctl enable apache2.service
[sudo] contraseña para fsuezarodriguez:
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
fsuezarodriguez@auriga:~$
```

Figura 2.24: Habilitación de Apache para que se ejecute al inicio

3. Respecto a la **configuración**, no hay mucho que comentar, ya que vamos a usar la que viene por defecto, porque no necesitamos ninguna configuración especial.
4. El siguiente punto es **copiar el archivo HTML** con nuestra página web y nuestra foto en el directorio **/var/www/html**. Además, cambiaremos el propietario a root, si no estuvieran ya así y los permisos a **755**, para que solo root pueda eliminar o modificar nuestra página web.



```
fsuezarodriguez@auriga:/var/www/html$ sudo cp ~/index.html .
fsuezarodriguez@auriga:/var/www/html$ sudo cp ~/Descargas/image.jpg .
fsuezarodriguez@auriga:/var/www/html$ sudo chown root.root *
fsuezarodriguez@auriga:/var/www/html$ sudo chmod 755 *
fsuezarodriguez@auriga:/var/www/html$ ls -las
total 36
4 drwxr-xr-x 2 root root 4096 may 21 12:55 .
4 drwxr-xr-x 3 root root 4096 may 21 12:15 ..
24 -rwxr-xr-x 1 root root 22642 may 21 12:55 image.jpg
4 -rwxr-xr-x 1 root root 498 may 21 12:55 index.html
fsuezarodriguez@auriga:/var/www/html$
```

Figura 2.25: Copia de index.html y cambio de permisos

5. Por último, vamos a **acceder a la página web** desde la **máquina anfitriona**. Como podemos ver, la página se ve correctamente, por lo que el servidor está funcionando a la perfección.



Figura 2.26: Acceso a la página web desde el anfitrión