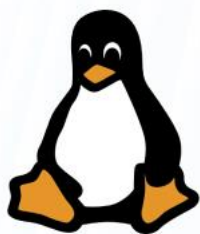


Sistemas Informáticos

Tarea N° 9



Manuel Pacheco Sánchez

Actividad 1. Configuración del entorno de red en Ubuntu 20.04.

Configura el equipo de Ubuntu 20.04 para que tenga siempre la misma dirección IP. Es decir, vamos a configurar de forma estática su dirección IP. Esta actividad será realizada usando los contenidos de la unidad como referencia.

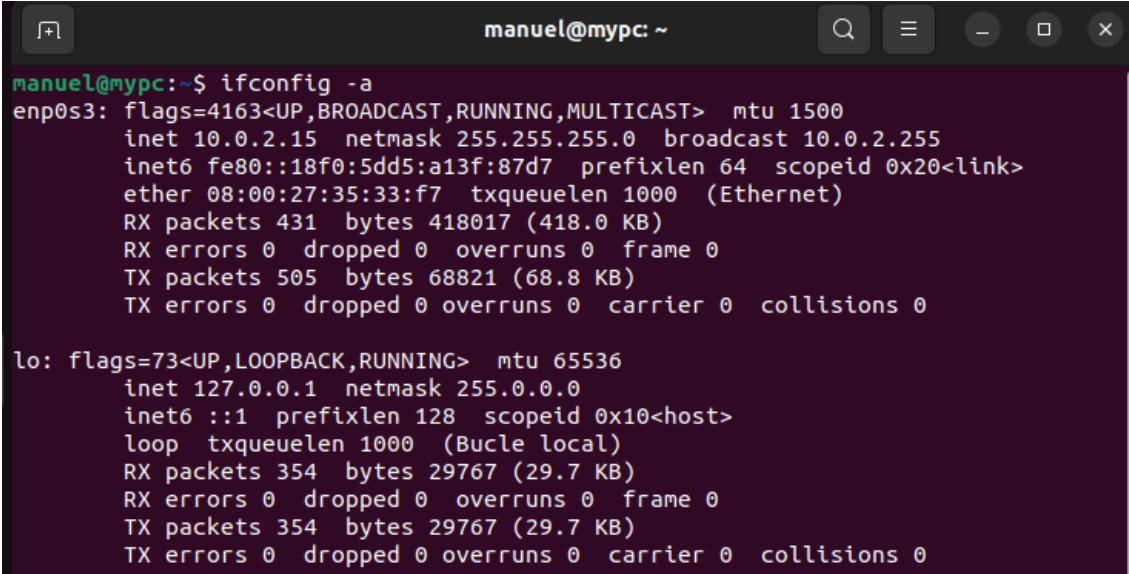
La dirección tiene que pertenecer a la red que estés usando actualmente. Si no trabajas con una red, es decir, no tienes una conexión a Internet por medio de un router en casa, toma como red de partida la siguiente: 192.168.5.0.

Una vez que hayas configurado la dirección IP realiza una captura de pantalla donde se vea su configuración desde el terminal de Ubuntu. Realiza un ping a la puerta de enlace y aporta una captura de pantalla donde se vea el resultado de su ejecución.

Como la actividad nos pide, vamos a configurar una dirección de manera estática en nuestra máquina virtual.

Primero, vamos a comprobar cual es la red en la que estamos, qué dirección tenemos y cual es la puerta de enlace a la que vamos a hacer ping.

Para comprobar nuestra ip, usamos **ifconfig -a**, instalando previamente **net-tools** para poder usarlo.



```
manuel@mypc:~$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
    inet6 fe80::18f0:5dd5:a13f:87d7  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:35:33:f7  txqueuelen 1000  (Ethernet)
    RX packets 431  bytes 418017 (418.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 505  bytes 68821 (68.8 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Bucle local)
    RX packets 354  bytes 29767 (29.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 354  bytes 29767 (29.7 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Como vemos, nuestra dirección es la 10.0.2.15. Vamos a asignar a nuestro equipo manualmente una dirección de la misma red.

Para ello, vamos a la configuración de red, al apartado de IPV4 y como método de conexión seleccionamos “manual”, e introducimos todos los parámetros necesarios.

Cancelar **Cableada** Aplicar

Detalles Identidad **IPv4** IPv6 Seguridad

Método IPv4 ☐ Automático (DHCP) ☐ Sólo enlace local
☒ Manual ☐ Desactivar
☐ Compartida con otros equipos

Direcciones

Dirección	Máscara de red	Puerta de enlace	
10.0.2.20	255.255.255.0	10.0.2.2	

DNS Automático ☒

Direcciones IP separadas por comas

La puerta de enlace la había consultado previamente a través del terminal, usando el comando **route -n**.

```
manuel@mypc: ~
manuel@mypc:~$ route -n
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic Métric Ref      Uso Interfaz
0.0.0.0      10.0.2.2      0.0.0.0      UG      100      0      0 enp0s3
10.0.2.0     0.0.0.0      255.255.255.0 U      100      0      0 enp0s3
169.254.0.0  0.0.0.0      255.255.0.0  U      1000     0      0 enp0s3
manuel@mypc:~$
```

La columna “Pasarela” nos devuelve la información de la puerta de enlace de nuestra red.

Vamos a reiniciar las interfaces de red para que reciba la configuración introducida manualmente. Podemos hacerlo o bien con el comando **sudo /etc/init.d/networking restart**, o bien reiniciando manualmente la conexión a internet desde los ajustes de red de Ubuntu.

Una vez reiniciamos las interfaces, vemos que nuestra IP ha cambiado a la que hemos introducido manualmente.

```
manuel@mypc: ~
manuel@mypc:~$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.20 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::18f0:5dd5:a13f:87d7 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:35:33:f7 txqueuelen 1000 (Ethernet)
    RX packets 469 bytes 422502 (422.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 591 bytes 78846 (78.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 396 bytes 33594 (33.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 396 bytes 33594 (33.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
manuel@mypc:~$
```

Ahora que tenemos una dirección IP estática, vamos a hacer ping a la puerta de enlace de la red, la cual se encuentra en la dirección **10.0.2.2**.

```
manuel@mypc: ~  
manuel@mypc:~$ ping 10.0.2.2  
PING 10.0.2.2 (10.0.2.2) 56(84) bytes of data.  
64 bytes from 10.0.2.2: icmp_seq=1 ttl=64 time=1.03 ms  
64 bytes from 10.0.2.2: icmp_seq=2 ttl=64 time=0.905 ms  
64 bytes from 10.0.2.2: icmp_seq=3 ttl=64 time=0.681 ms  
64 bytes from 10.0.2.2: icmp_seq=4 ttl=64 time=0.389 ms  
64 bytes from 10.0.2.2: icmp_seq=5 ttl=64 time=0.406 ms  
64 bytes from 10.0.2.2: icmp_seq=6 ttl=64 time=0.692 ms  
64 bytes from 10.0.2.2: icmp_seq=7 ttl=64 time=0.347 ms  
64 bytes from 10.0.2.2: icmp_seq=8 ttl=64 time=0.898 ms  
64 bytes from 10.0.2.2: icmp_seq=9 ttl=64 time=1.25 ms  
  
--- 10.0.2.2 ping statistics ---  
9 packets transmitted, 9 received, 0% packet loss, time 8087ms  
rtt min/avg/max/mdev = 0.347/0.732/1.247/0.295 ms  
manuel@mypc:~$
```

Como vemos, la puerta de enlace recibe los paquetes enviados desde nuestro equipo.

Actividad 2. Compartimos recursos en la red.

Instala en la MV lo que sea necesario para que Ubuntu pueda ver/editar una carpeta compartida con el SO anfitrión estando en la misma red. En el SO anfitrión se creará una carpeta compartida. El objetivo es poder acceder desde Ubuntu a dicha carpeta compartida, para crear un fichero y leerlo.

Ten en cuenta que según sea tu sistema anfitrión es posible que necesites instalar en él o en la máquina virtual distintos paquetes (Samba, NFS, APFS, etc.).

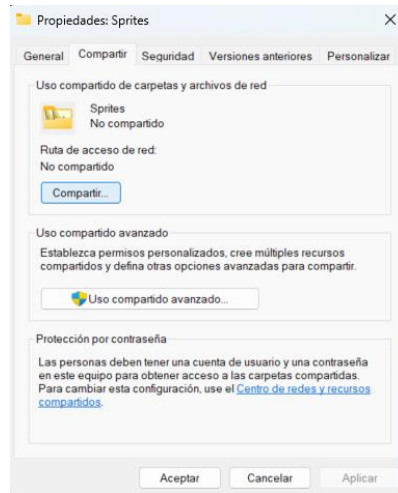
Para compartir una carpeta desde nuestro Host Windows a nuestra máquina virtual Ubuntu, tenemos que añadir una interfaz de red a nuestro Ubuntu que funcione como adaptador puente entre ambos equipos. Para ello, desde la configuración de nuestra máquina virtual, añadimos una nueva interfaz y especificamos que funcione como adaptador puente.

Una vez entramos en Ubuntu, vamos a las configuraciones de red, y en la nueva interfaz, introducimos manualmente la dirección IPV4 que queramos siempre que se encuentre en el mismo rango de direcciones que la de nuestro Host Windows. En mi caso, he asignado la dirección **192.168.0.45**. Es conveniente probar que tenemos conexión entre ambas máquinas haciendo ping.

```
C:\Users\Manuel Pacheco>ping 192.168.0.45  
  
Haciendo ping a 192.168.0.45 con 32 bytes de datos:  
Respuesta desde 192.168.0.45: bytes=32 tiempo<1m TTL=64  
Respuesta desde 192.168.0.45: bytes=32 tiempo<1m TTL=64  
Respuesta desde 192.168.0.45: bytes=32 tiempo<1m TTL=64  
Respuesta desde 192.168.0.45: bytes=32 tiempo<1m TTL=64  
  
Estadísticas de ping para 192.168.0.45:  
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0  
    (0% perdidos),  
    Tiempos aproximados de ida y vuelta en milisegundos:  
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms  
  
C:\Users\Manuel Pacheco>
```

Para gestionar la conexión al directorio compartido desde Windows, vamos a instalar en Ubuntu **Samba** usando **apt install samba**.

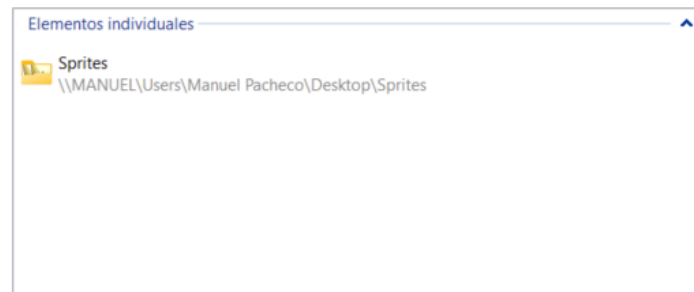
Para compartir el directorio, vamos a sus propiedades, a la pestaña de “Compartir” y pulsamos sobre compartir.



Damos permisos a todos los usuarios para poder leer y escribir en este directorio y pulsamos en aceptar.

La carpeta está compartida.

Puedes [enviar por correo electrónico](#) a cualquier persona vínculos a estos elementos compartidos o [copiar](#) los vínculos y pegarlos en otra aplicación.



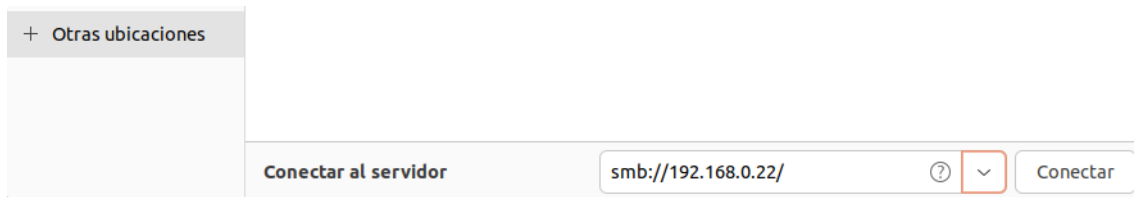
[Mostrar todos los recursos compartidos de red en este equipo.](#)

Listo

Ahora que nuestra carpeta está compartida en la red, vamos a conectarnos desde Ubuntu.

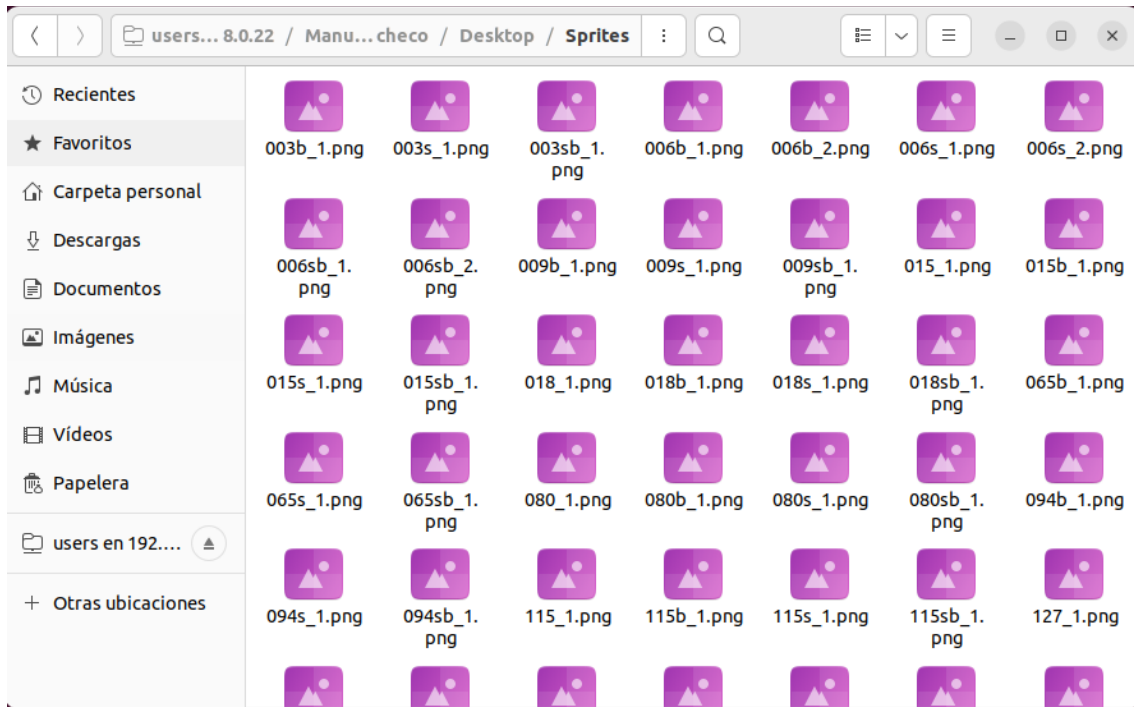
Para ello, vamos al explorador de archivos, y en la barra de navegación izquierda, vamos a “Otras ubicaciones”.

En la barra inferior tenemos una caja de texto, donde debemos introducir la dirección IP del equipo al que vamos a conectarnos, el cual tiene la carpeta compartida, y el cual, en nuestro caso, es la dirección del host Windows.

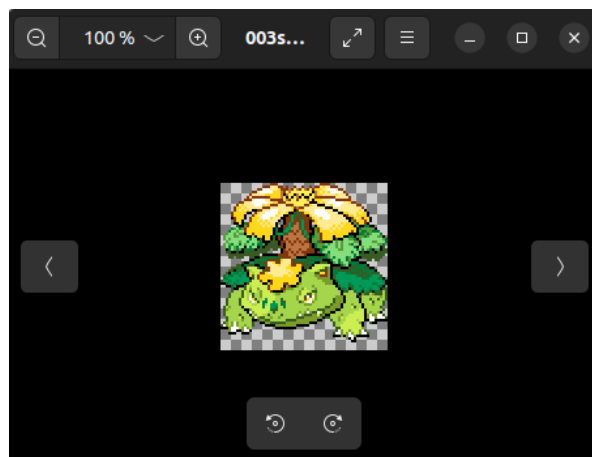


Introducimos nuestro nombre de usuario y contraseña de Windows, y pulsamos en Conectar.

Una vez nuestras credenciales son correctas, tendremos acceso al sistema de archivos de nuestro equipo Windows desde Ubuntu.

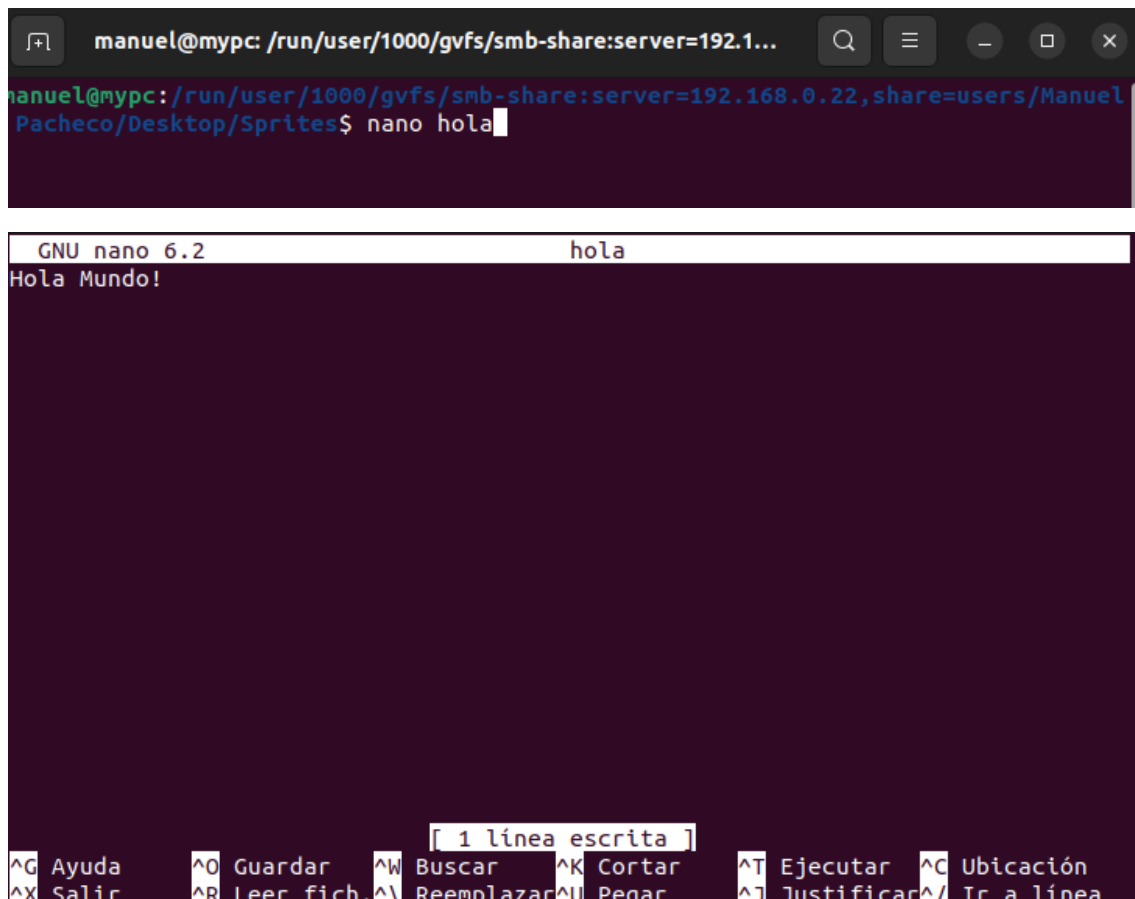


Como vemos, tenemos acceso a la carpeta que hemos compartido previamente.

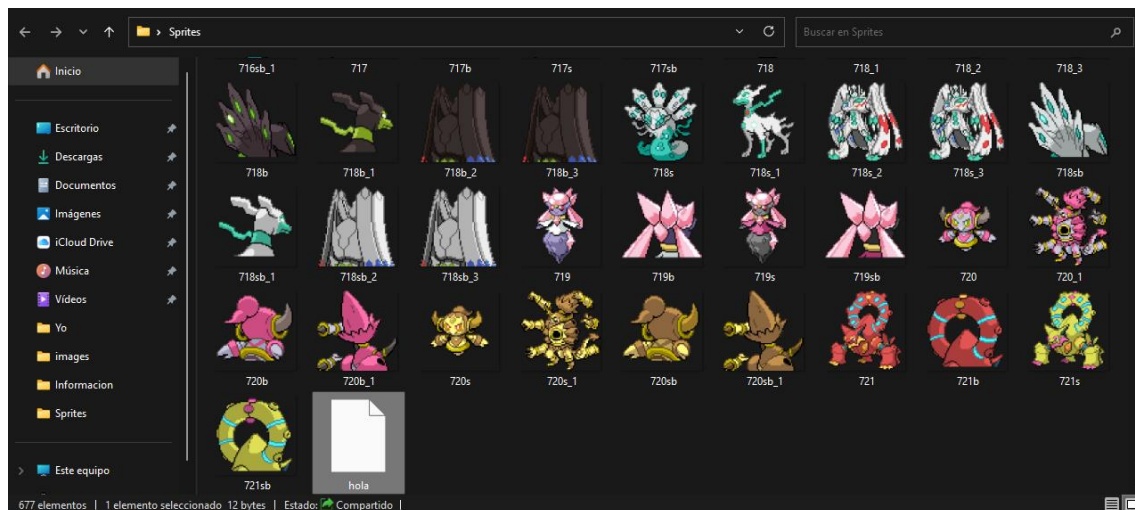


Vemos que tenemos acceso a la lectura de archivos.

Vamos a abrir este directorio desde el terminal y crear un archivo de texto usando **nano** para comprobar que tenemos también permisos de escritura.



Una vez guardamos el archivo, al volver a acceder al directorio, debe existir tanto en Windows como en Ubuntu. Vamos a buscar el fichero en Windows.



Efectivamente, nuestro fichero es visible desde nuestro host Windows.

Actividad 3. Conexión remota a dispositivos.

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

- SSH
- VNC

Para conectarnos mediante SSH, primero tenemos que instalar el servicio en Ubuntu.

```
root@mypc: /
root@mypc:/# apt-get install ssh
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
Paquetes sugeridos:
  molly-guard monkeysphere ssh-askpass
Se instalarán los siguientes paquetes NUEVOS:
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 55 no actualizados.
Se necesita descargar 755 kB de archivos.
Se utilizarán 6.179 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

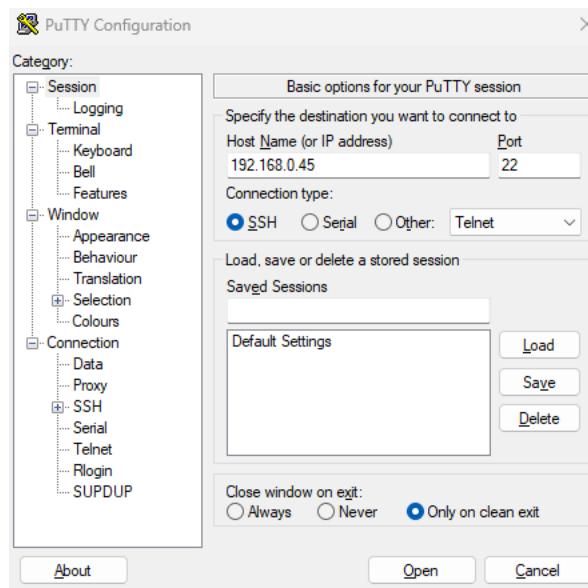
Una vez instalado, lo iniciamos.

```
root@mypc:/# service ssh start
root@mypc:/# service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en
   Active: active (running) since Thu 2023-05-04 16:37:56 CEST; 49s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 3759 (sshd)
      Tasks: 1 (limit: 4614)
     Memory: 1.7M
        CPU: 30ms
    CGroup: /system.slice/ssh.service
            └─3759 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

may 04 16:37:56 mypc systemd[1]: Starting OpenBSD Secure Shell server...
may 04 16:37:56 mypc sshd[3759]: Server listening on 0.0.0.0 port 22.
may 04 16:37:56 mypc sshd[3759]: Server listening on :: port 22.
may 04 16:37:56 mypc systemd[1]: Started OpenBSD Secure Shell server.
lines 1-16/16 (END)
```

Una vez iniciado SSH en Ubuntu, vamos a instalar PuTTY, un cliente SSH, en nuestro host Windows.

Teniendo instalado PuTTY, lo abrimos e introducimos la dirección a la que queremos conectarnos, la cual va a ser la IP de nuestra máquina virtual.



Al entrar, nos preguntará con que usuario queremos acceder, nos solicita la contraseña del mismo, y al introducir las credenciales correctamente, tendremos completo acceso al terminal de nuestro Ubuntu.

```
manuel@mypc:~$ ls
Descargas  Escritorio  Música      Público  Videos
Documentos Imágenes    Plantillas  snap
manuel@mypc:~$ cd Escritorio
manuel@mypc:~/Escritorio$ ls
scripts
manuel@mypc:~/Escritorio$
```

Para conectarnos a través de VNC, primero tenemos que instalar **tigerVNC** en nuestro equipo.

```
root@mypc:/# apt install tigervnc-standalone-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libfile-readbackwards-perl tigervnc-common tigervnc-tools
Paquetes sugeridos:
  xfonts-100dpi | xfonts-75dpi
Se instalarán los siguientes paquetes NUEVOS:
  libfile-readbackwards-perl tigervnc-common tigervnc-standalone-server
  tigervnc-tools
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 55 no actualizados.
Se necesita descargar 1.272 kB de archivos.
Se utilizarán 3.305 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 libfile-readbackwards-perl all 1.06-1 [11,2 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 tigervnc-common amd64 1.12.0+dfsg-4 [101 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 tigervnc-standalone-server amd64 1.12.0+dfsg-4 [1.138 kB]
75% [3 tigervnc-standalone-server 926 kB/1.138 kB 81%]
```

Establecemos una contraseña de acceso a nuestro servidor VNC:

```
root@mypc:/# vncpasswd
Password:
Verify:
Would you like to enter a view-only password (y/n)? n
A view-only password is not used
root@mypc:/#
```

Por último, en nuestra máquina Ubuntu ejecutamos el comando **vncserver** para inicializarlo, lo que nos devolverá el puerto donde se ha iniciado.

```
root@mypc:/# vncserver

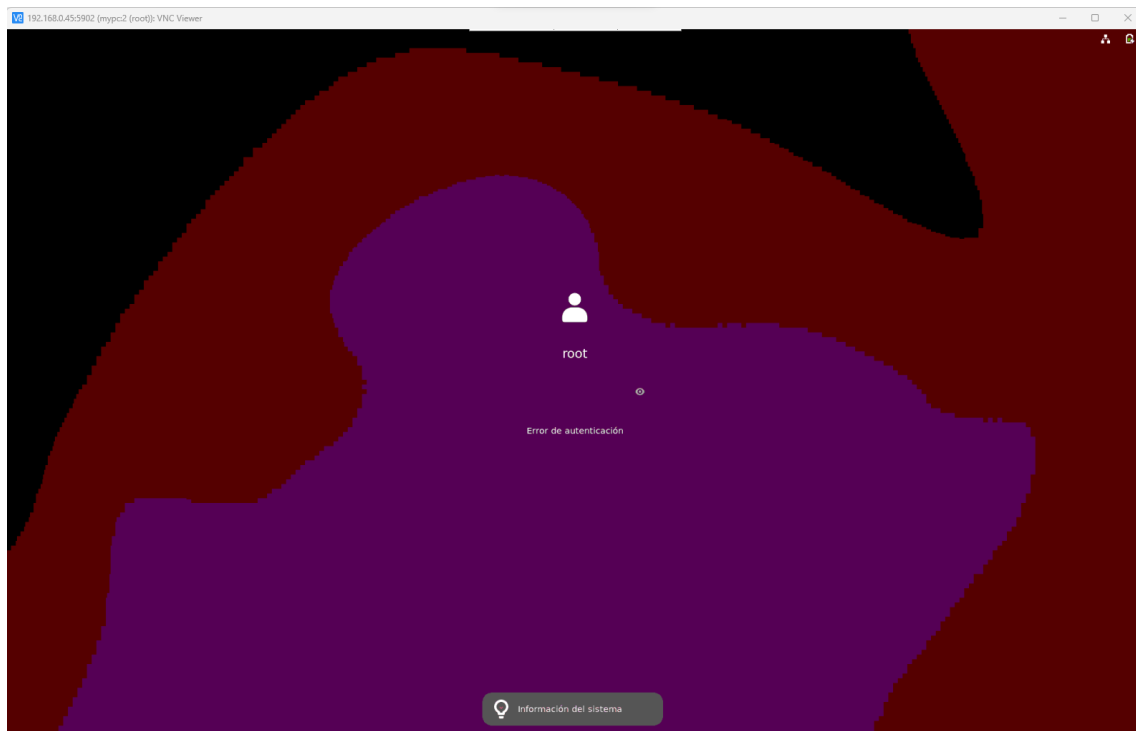
New Xtigervnc server 'mypc:2 (root)' on port 5902 for display :2.
Use xtigervncviewer -SecurityTypes VncAuth -passwd /root/.vnc/passwd :2 to connect to the VNC server.

root@mypc:/#
```

Tenemos que introducir el comando **vncserver -localhost no** para que acepte conexiones externas al localhost.

Una vez tenemos inicializado el servidor VNC, vamos a instalar en Windows VNC Viewer, que será el cliente con el que nos conectaremos al servidor VNC.

Entramos en VNC Viewer, e introducimos la IP de nuestra máquina virtual, seguido del puerto, donde tenemos abierto el servidor.



Como vemos, podemos acceder desde nuestro Windows a la máquina virtual con Ubuntu.

Actividad 4. Instalación de servicios. Servicio FTP.

En esta actividad se instalará en Ubuntu un servidor FTP y en el SO anfitrión un cliente FTP como Filezilla para acceder desde él a nuestro servidor instalado.

Una vez que esté todo instalado realiza una transmisión de algún archivo del servidor FTP al cliente y viceversa.

Para crear un servidor FTP en nuestro Ubuntu, tenemos que instalar **vsftpd** mediante el comando **sudo apt install vsftpd**.

```
root@mypc:~# apt install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  vsftpd
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 123 kB de archivos.
```

Vamos a acceder al archivo de configuración de vsftpd, ubicado en **etc/vsftpd.conf**.

Vamos a permitir el acceso a usuarios no registrados:

```
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
```

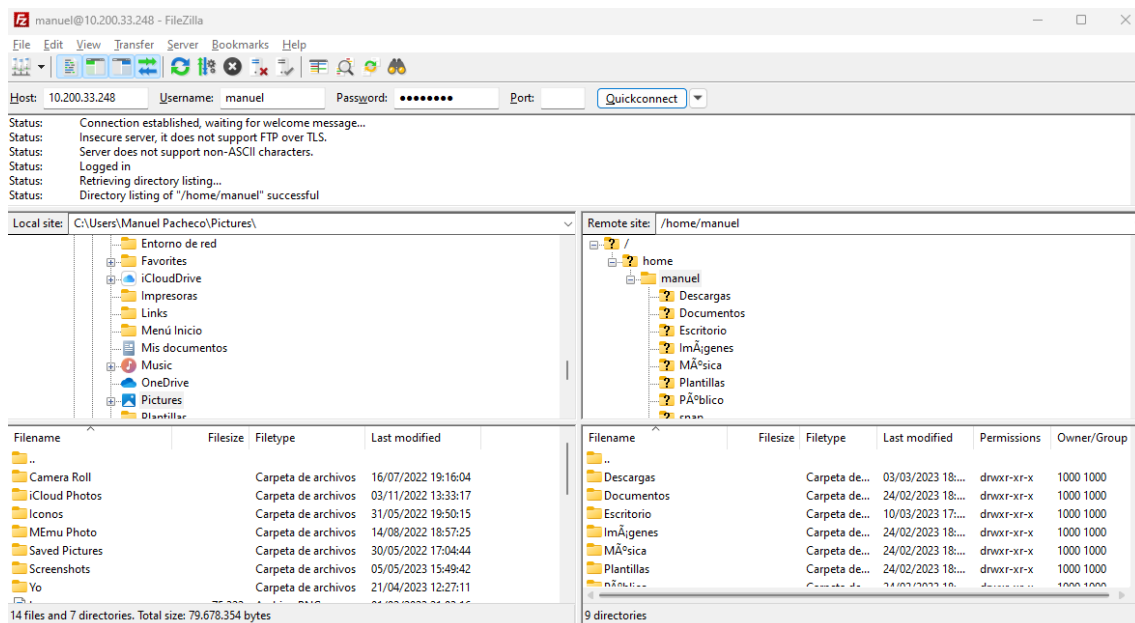
También vamos a descomentar la línea que permite la escritura de ficheros:

```
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
```

Vamos a reiniciar el servidor FTP para que se apliquen los cambios en la configuración.

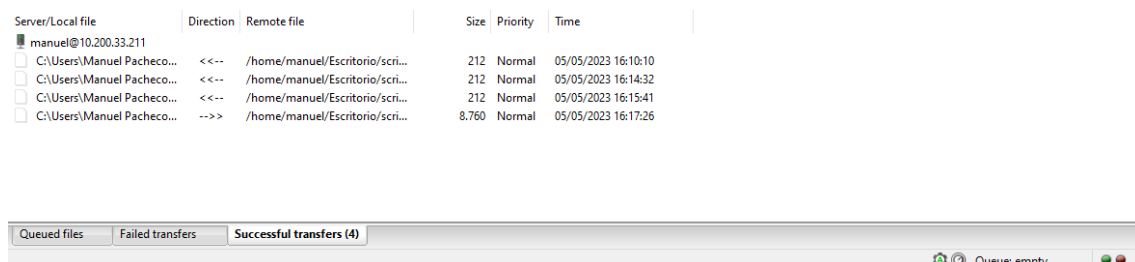
```
root@mypc: /etc
root@mypc:/etc# systemctl restart vsftpd
root@mypc:/etc#
```

Ahora, vamos a intentar conectarnos a nuestro servidor desde el host Windows, a través del puerto 21.



Introduciendo la dirección IP de Ubuntu, nuestro nombre de usuario y contraseña, y el puerto 21, tenemos conexión correctamente con nuestro servidor FTP y podemos ver el directorio raíz del usuario “Manuel”.

Vamos a intentar hacer varias transferencias, tanto de descarga como de subida de archivos.



Como vemos, las transferencias se realizan de manera exitosa.

Actividad 5. Instalación de servicios. Servidor Web.

Instala Apache en el equipo de Ubuntu, configúralo para que se inicie automáticamente y abre en un navegador desde el equipo anfitrión la página web que creaste en actividades anteriores.

Si no realizaste la mencionada actividad donde creaste tu página web, a continuación se detalla el código de la página web que debes crear y guardar en el archivo mipagina.html. Para ello utiliza un editor de texto en Ubuntu y en la imagen no olvides poner tu foto.

Antes que nada, vamos a instalar **apache2**, que será el que nos permita levantar nuestro servidor Apache.

```

root@mypc:/# apt remove 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.
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Utilice «sudo apt autoremove» para eliminarlos.
Los siguientes paquetes se ELIMINARÁN:
  apache2
0 actualizados, 0 nuevos se instalarán, 1 para eliminar y 1 no actualizados.
Se liberarán 546 kB después de esta operación.
¿Desea continuar? [S/n] S
(Leyendo la base de datos ... 266368 ficheros o directorios instalados actualmen
te.)

```

Vamos a comprobar que se ha iniciado el servidor Apache correctamente.

```

root@mypc:/# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Fri 2023-05-05 16:27:24 CEST; 1min 9s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 4034 (apache2)
      Tasks: 55 (limit: 4614)
     Memory: 4.8M
        CPU: 64ms
    CGroup: /system.slice/apache2.service
            └─4034 /usr/sbin/apache2 -k start
              └─4035 /usr/sbin/apache2 -k start
                └─4036 /usr/sbin/apache2 -k start

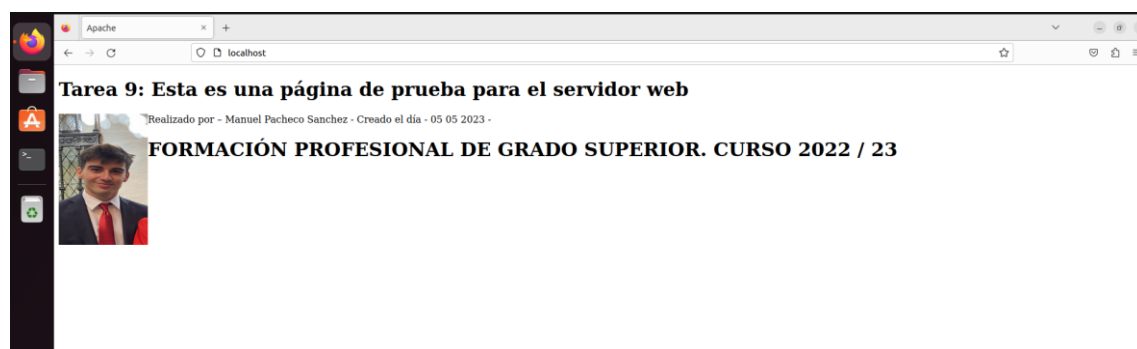
may 05 16:27:24 mypc systemd[1]: Starting The Apache HTTP Server...
may 05 16:27:24 mypc apachectl[4033]: AH00558: apache2: Could not reliably dete>
may 05 16:27:24 mypc systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)

root@mypc:/#

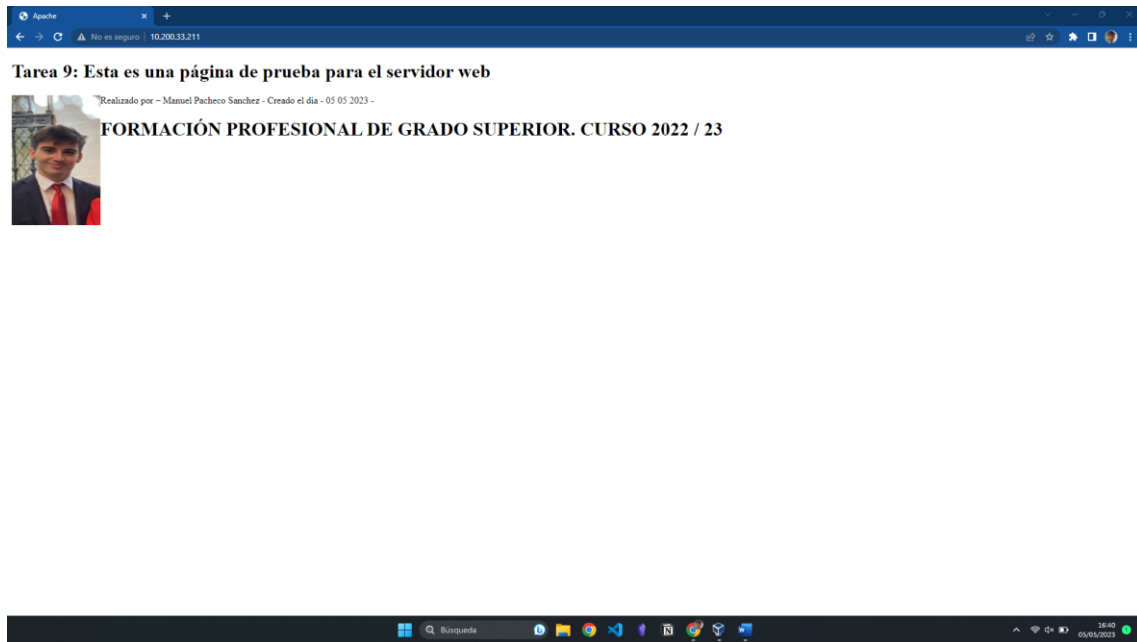
```

Una vez nuestro servidor está en correcto funcionamiento, vamos a acceder al directorio **var/www/html** y vamos a modificar el fichero html para que cumpla con el que nos pone en la actividad.

Una vez lo tengamos listo, vamos a acceder desde el navegador de nuestro Ubuntu a localhost para visualizar la web que hemos creado.



Y ahora vamos a acceder desde Windows a la dirección IP de nuestra máquina virtual de Ubuntu, debiéndonos mostrar igualmente la web que hemos creado.



Como vemos, también se muestra en Windows, por lo que nuestro servidor es completamente operativo.