

Caso práctico

María visita a Juan.

-Juan, vamos a mejorar la red de la empresa y para eso necesito que utilices un servidor para que actúe como router y le proporcione a la empresa los servicios más importantes. Encaminamiento y DHCP, y luego le instalaremos más servicios.



-Muy bien María, he visto por Internet que Linux funciona muy bien con redes. De hecho su máxima potencia se utiliza para actuar como servidor. Ahora mismo me pongo a configurar el servidor.



[Ministerio de Educación y Formación Profesional](#). (Dominio público)

Materiales formativos de FP Online propiedad del Ministerio de Educación y Formación Profesional.

[Aviso Legal](#)

1.- Configuración de red y router en Linux.

La unidad comienza con la configuración de 2 máquinas Linux en red; se configurarán en la misma red interna que las dos máquinas de Windows de la unidad 9.

En la primera máquina Linux se pondrán dos tarjetas de red, una en NAT con la máquina anfitrión por la que sale a Internet, y la otra en red interna con las otras 3 máquinas. Esta máquina Linux realizará las funciones de router, de forma que las otras 3 máquinas también saldrán a Internet a través de ésta.

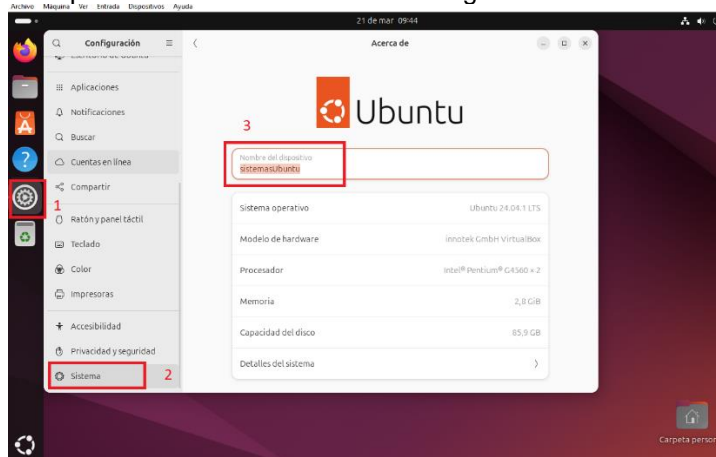
En el resto de la unidad, se estudian los distintos servicios de forma que en esta máquina router se instalarán distintos servicios: samba, nfs, ssh,...

1.1.- Configuración de 2 máquinas Linux en Red.

Este apartado es el primer ejercicio de la tarea de la unidad.

Paso 1. Clonar la máquina Linux utilizada en anteriores unidades

1. Clonar con VirtualBox la máquina Linux utilizada en unidades anteriores. Realizar una clonación completa y reiniciando la dirección MAC.
2. En la máquina clonada, poner como nombre de máquina **clienteLinux**.
Para hacerlo tenemos dos opciones: o bien desde el entorno gráfico:



Ana Belén Ocoń Pérez de Albéniz (CC BY-NC-SA)

O bien empleando el comando `hostnamectl`. Escribiéndolo solo se mostrará el nombre del equipo, además ofrece otros datos, como el sistema operativo, la versión del núcleo, la arquitectura, etc. En nuestro ejemplo, incluso detecta que el sistema está virtualizado y nos dice con qué herramienta.

Para cambiar el nombre deberemos añadir un parámetro al comando:
`sudo hostnamectl set-hostname nuevo-nombre`

Esto parece todo pero aún nos queda un pequeño detalle:

3. Editar el archivo `/etc/hosts` y cambiar el nombre de la máquina (donde aparece `SistemasUbuntu`, cambiarlo por `clienteLinux`)
Observación: el archivo `/etc/hosts` sirve para configurar un DNS básico en la red local.

Paso 2. Configuración de red en máquina Linux servidor.

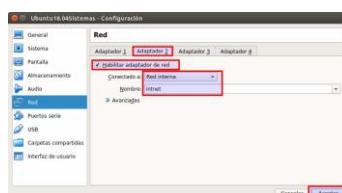
La máquina original Linux va a tener 2 tarjetas de red para configurar un router:

- ✓ La primera tarjeta de red, sigue como hasta ahora, en NAT con la máquina anfitrión, saliendo a Internet a través de ella.
- ✓ La segunda tarjeta de red, la pondremos en red interna para comunicarse con la otra máquina Linux.

1. Añadir segunda tarjeta de red a la máquina

En VirtualBox en configuración de Red, habilitar adaptador 2 (pestaña adaptador 2) y ponerlo en red interna, tal como aparece en la imagen.

Con esta operación, esta máquina ya se encuentra en la misma red física que las 2 máquinas Windows, (están en el mismo switch), pues el nombre de la red interna no lo hemos cambiado. Cuando configuremos la IP se configurará de forma que estén en la misma red lógica.



2. Nombres de las tarjetas de red en Linux. Comando ifconfig

Al ejecutar ifconfig se ven los nombres asignados por Ubuntu a las tarjetas de red y sus direcciones IP asignadas. En la captura que hay a continuación se visualiza:

- ✓ Tarjeta enp0s3 (tarjeta de red en NAT con maquina anfitrión) con la dirección IP 10.0.2.15
- ✓ Tarjeta enp0s8 (tarjeta de red en red interna, que comunicará con todas las máquinas restantes) sin dirección IP de momento.

```
miguel@SistemasUbuntu: ~
Archivo Editar Ver Buscar Terminal Ayuda
miguel@SistemasUbuntu:~$ ifconfig
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::8fe6:b274:f65b:9eec prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:ff:a4:27 txqueuelen 1000 (Ethernet)
    RX packets 1197 bytes 1352426 (1.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 599 bytes 62488 (62.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::81c3:3f47:9cd4:3ff9 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:43:9a:4b txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 216 bytes 35581 (35.5 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
```

Miguel Ángel García Lara (CC BY-NC-SA)

Con el comando ip address también obtenemos el mismo resultado:

```
estudiante@sistemasubuntu:~$ ip address
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:c5:61:51 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 84808sec preferred_lft 84808sec
    inet6 fd00::aef:25e7:b552:cbc0/64 scope global temporary dynamic
        valid_lft 86219sec preferred_lft 14219sec
    inet6 fd00::a00:27ff:fec5:6151/64 scope global dynamic mngtppaddr
        valid_lft 86219sec preferred_lft 14219sec
    inet6 fe80::a00:27ff:fec5:6151/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:3c:f1:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.103/24 brd 192.168.100.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::ddb4:72fc:1b3d:daae/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
estudiante@sistemasubuntu:~$
```

Ana Belén Ocoń Pérez de Albéniz (CC BY-NC-SA)

Con el comando route -n vemos que puerta de enlace nos asigna VirtualBox en el adaptador de red NAT para salir a Internet:

```
estudiante@sistemasubuntu:~$ 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
192.168.100.0 0.0.0.0      255.255.255.0 U      101     0      0 enp0s8
estudiante@sistemasubuntu:~$
```

Ana Belén Ocoń Pérez de Albéniz (CC BY-NC-SA)

Observación: el comando ifconfig puede no estar instalado

La primera vez que se ejecuta `ifconfig`, en Ubuntu 24.04 dice que no está instalado y que hay que instalar `net-tools`:

```
ana@sistemasUbuntu:~$ ifconfig
```

No se ha encontrado la orden «ifconfig», pero se puede instalar con:

```
sudo apt install net-tools
```

```
ana@sistemasUbuntu:~$ sudo apt install net-tools
```

```
....
```

Configurando net-tools (1.60+git20161116.90da8a0-1ubuntu1)

```
ana@sistemasUbuntu:~$
```

Es posible, que incluso de algún error al instalar `net-tools`, porque no se haya actualizado nada en nuestro Ubuntu. En ese caso será necesario actualizar los paquetes instalados previamente. Ejecutar los 2 comandos siguientes:

- Actualizar lista paquetes: `#apt update`
- Actualizar paquetes instalados: `#apt upgrade`

En Ubuntu 24 podemos ver qué adaptadores de red tenemos empleando:

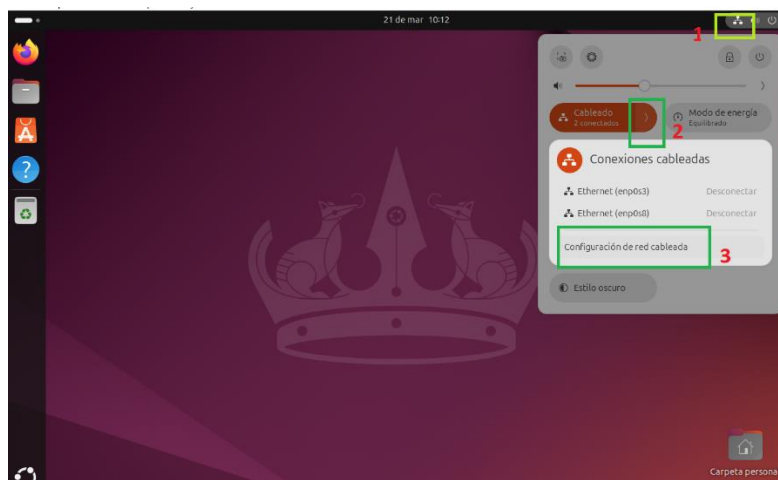
```
ana@sistemasUbuntu:~$ ls /sys/class/net
```

```
enp0s3 enp0s8 lo
```

3. Configurar IP de la segunda tarjeta de red

La configuración de las tarjetas de red en Ubuntu se encuentra en el directorio `/etc/netplan`.

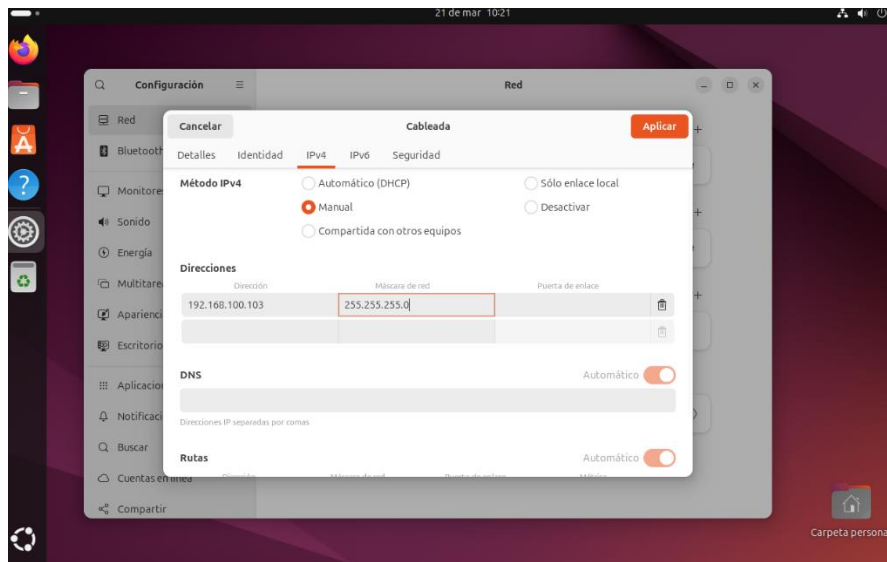
Lo haremos desde el entorno gráfico:



Ana Belén Ocoń Pérez de Albéniz (CC BY-NC-SA)

La configuración del primer adaptador de red (`enp0s3`) lo dejaremos como está: Automático (DHCP), el propio VirtualBox le asignará una dirección de la red 10.0.2.0/24.

Sí modificaremos la configuración del segundo adaptador de red (el que comunica con la red interna):



Ana Belén Ocón Pérez de Albéniz (CC BY-NC-SA)

Reiniciar la red y comprobar que funciona enps08

Para que tenga efecto el cambio, se reinicia el servicio de red.

```
root@SistemasUbuntu:~# systemctl restart systemd-networkd
```

```
root@SistemasUbuntu:~# ifconfig #vemos que enp0s8 tiene la nueva dirección IP asignada
```

```
root@SistemasUbuntu:~# ping 192.168.100.103
```

#debe responder afirmativamente comprobando que hay conectividad de red en el propio equipo.

```
root@SistemasUbuntu:~# ping 8.8.8.8
```

#responde pues, tenemos conexión a Internet y esta dirección IP existe. Esta dirección IP corresponde al servidor DNS de Google.

Paso 3. Configuración en máquina clienteLinux

Esta máquina solo tiene una tarjeta de red. Configurar en VirtualBox la tarjeta en red interna.

Al ejecutar ifconfig se observa que solo hay una interfaz de red con el nombre enps03. Se configura la dirección igualmente desde el entorno gráfico asignado una IP fija 192.168.100.104 y la puerta de enlace 192.168.100.103

Recordar que la puerta de enlace o gateway es la dirección IP por la que nos comunicamos con otras redes. En la primera máquina Linux no había que añadir puerta de enlace, pues sale a Internet directamente a través del apartador de red NAT. Pero esta segunda máquina Linux, saldrá a Internet a través de la primera, configurando como puerta de enlace su dirección IP.

Para que tenga efecto el cambio, se reinicia el servicio de red:

```
root@clienteLinux:~# systemctl restart systemd-networkd
```

En el siguiente apartado, se configurará la primera máquina para que sea un router; de forma que todas estas máquinas puedan salir a Internet a través de la máquina SistemasUbuntu.

Comprobaciones de que la nueva conexión funciona:

```
root@clienteLinux:~# ifconfig #vemos que enp0s3 funciona con nueva ip
```

```
root@clienteLinux:~# ping 192.168.100.104
```

#nuestra propia ip responde al ping.

```
root@clienteLinux:~# ping 192.168.100.103 #responde la otra máquina linux
```

```
root@clienteLinux:~# ping 8.8.8.8 #no responde, pues aunque se haya configurado la puerta de enlace, la primera máquina aún no está configurada como router, por lo que no comunica las 2 redes.
```

Observación

Ahora hay 4 máquinas en la misma red:

Las 2 máquinas Windows con IP 192.168.100.101 y 192.168.100.102

Las 2 máquinas Linux con IP 192.168.100.103 y 192.168.100.104

Todas con la máscara 255.255.255.0

Para saber más

Para ver más información sobre configurar red en Ubuntu 18.04 ver vínculo: [Configurar ip estática en Ubuntu 18.04](#)

1.2.- Enrutamiento en Linux.

En este apartado se va a habilitar el enrutamiento en la máquina SistemasUbuntu. Se activará regla iptables para permitir el tráfico, de forma que el resto de las máquinas salgan a Internet a través del router SistemasUbuntu.

Este contenido es el ejercicio 2 de la tarea.

Paso 1. Habilitar router en máquina SistemasUbuntu.

En **primer lugar**, se **activa el servicio de enrutamiento** de Linux. Para ello se activa ip_forward de la forma siguiente:

```
root@SistemasUbuntu:~# cat /proc/sys/net/ipv4/ip_forward
```

```
0
```

```
root@SistemasUbuntu:~# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Con el comando echo, hemos sustituido un 0 por 1 dentro del archivo. De esta forma, decimos a la máquina que va a enrutar.

En **segundo lugar** se **activa una regla iptable**, servicio de cortafuegos de Linux, que diga que deja pasar todo el tráfico.

```
root@SistemasUbuntu:~# iptables -t nat -A POSTROUTING -s 192.168.100.0/24 -d 0/0 -j MASQUERADE
```

Este comando permite que los dispositivos dentro de la red 192.168.100.0/24 salgan a internet utilizando la dirección IP del servidor como dirección de origen, permitiendo que el tráfico se "enmascare", lo que es útil en configuraciones NAT.

Si quisiéramos restricciones de tráfico, tendríamos que habilitar más reglas iptables.

Paso 2. Comprobar enrutamiento en máquina clienteLinux

Para comprobar que la máquina SistemasUbuntu ya está enrutando, se ejecuta ping en la máquina clienteLinux hacia el exterior:

```
root@clienteLinux:~# ping 8.8.8.8 #Responde, ya hay comunicación con Internet
```

```
root@clienteLinux:~# ping www.diariodenavarra.es #No responde, pues no se ha configurado la dirección IP del servidor DNS.
```

#No hay ningún equipo en nuestra red que traduzca www.diariodenavarra.es a la IP concreta.

Configurar DNS en máquinas Linux

Cuando se utilizan direcciones IP estáticas, además de la máscara de red, se debe configurar la puerta de enlace y los servidores DNS a utilizar. En este caso, se ha decidido configurar como DNS los propios servidores DNS de Google con direcciones IP 8.8.8.8 (DNS primario) e IP 8.8.4.4 (DNS secundario). Siempre se configuran 2 servidores DNS por si el primero falla en la conexión.

Para realizarlo, ejecutar:

```
root@clienteLinux:~# mv /etc/resolv.conf /etc/resolv.conf.old
```

#Guardamos el archivo resolv.conf original con otro nombre, por si en un futuro queremos tener el archivo original. A continuación se insertan los 2 DNS en el archivo.

```
root@clienteLinux:~# echo nameserver 8.8.8.8 > /etc/resolv.conf
```

```
root@clienteLinux:~# echo nameserver 8.8.4.4 >> /etc/resolv.conf
```

```
root@clienteLinux:~# cat /etc/resolv.conf
```

```
nameserver 8.8.8.8
```

```
nameserver 8.8.4.4
```

```
root@clienteLinux:~# ping www.diariodenavarra.es #ya responde afirmativamente, pues el servidor DNS sabe que IP tiene el servidor www.diariodenavarra.es
```

Paso 3. Realizar un script con inicio automático, para que el enrutamiento se inicie siempre.

Si reiniciamos la máquina SistemasUbuntu, ya no enrutará. Es decir, la máquina clienteLinux ya no responde afirmativamente a ping 8.8.8.8.

¿Cómo solucionarlo?

Crear un script con los comandos del paso 1. Este script se ejecutará siempre que se inicie el equipo de forma

automática.

En Ubuntu el archivo **/etc/rc.local** se ejecuta siempre que se inicia GNU-Linux.

En Ubuntu 18.04 este archivo no existe, pero lo creamos e introducimos los comandos vistos:

```
root@SistemasUbuntu:~# nano /etc/rc.local
```

#Se escriben las 4 líneas siguientes:

```
#!/bin/bash
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
iptables -t nat -A POSTROUTING -s 192.168.100.0/24 -d 0/0 -j MASQUERADE
```

```
exit 0 #Siempre debe ser la última línea del script rc.local
```

```
root@SistemasUbuntu:~# chmod +x /etc/rc.local
```

#Se cambia permisos al archivo para que sea ejecutable.

```
root@SistemasUbuntu:~# reboot
```

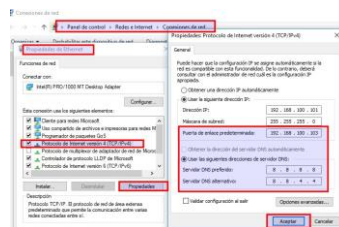
#Al reiniciar el equipo, se ejecuta rc.local, de forma que esta máquina ya enruta en todo momento. Comprobar que clienteLinux responde bien a ping www.diariodenavarra.es

Paso 4. Salida a Internet de máquinas Windows cliente1 y cliente2

Para que las máquinas Windows de la unidad 9 salgan a Internet, solo falta configurar en ellas la puerta de enlace y el servidor DNS que en su momento los dejamos en blanco según captura.

Configurar como puerta de enlace la máquina SistemasUbuntu: 192.168.100.103

Configurar como DNS los servidores de Google, 8.8.8.8 y 8.8.4.4.



Miguel Ángel García Lara (CC BY-NC-SA)

Tener en cuenta que según la memoria RAM de la máquina anfitrión es posible que solo se puedan tener 2 máquinas virtuales encendidas

En esta unidad, para realizar las prácticas siempre tendrá que estar encendida la máquina SistemasUbuntu, y una más para ir haciendo las comprobaciones necesarias.

1.3.- Servicios y comandos TCP/IP en GNU-Linux.

Servicios en GNU-Linux

Los servicios en Linux se configuran con un archivo de texto, normalmente con extensión "conf". Así, una vez instalado el servicio web Apache se configura en el archivo /etc/apache2/apache2.conf

Los servicios en Linux se pueden parar e iniciar con el comando **systemctl**. Las cuatro opciones más usadas: **stop**, **start**, **restart**, **status**.

Cada vez que se modifica la configuración de un servicio deberemos reiniciarlo para que las modificaciones tengan efecto.

Si son varios los servicios que queremos reiniciar, puede que sea más rápido reiniciar la máquina por completo.

Como ejemplo, para el servicio **systemd-networkd** (servicio de red) se puede ejecutar:

```
# systemctl start systemd-networkd  #iniciar servicio de red
# systemctl stop systemd-networkd   #parar servicio de red
# systemctl restart systemd-networkd #reiniciar servicio de red
# systemctl status systemd-networkd  #informa si el servicio se está ejecutando o está parado. Cuando se
está ejecutando, aparece la palabra running (en color verde).
```

Comandos TCP/IP en Linux

Los comandos TCP/IP como dicen su nombre, no son exclusivos de Windows o Linux, sino del protocolo TCP. De ahí que los comandos realizan la misma acción y sólo cambia el nombre alguno ligeramente. La relación en Linux es:

- ✓ ifconfig -a (equivale a ipconfig en Windows)
- ✓ ping
- ✓ hostname
- ✓ arp
- ✓ traceroute (equivale a tracert en Windows)
- ✓ netstat
- ✓ nslookup
- ✓ route

2.- Servicio SAMBA.

El servicio SAMBA sirve para compartir recursos entre máquinas Windows y Linux, para ello se utiliza el protocolo SMB, que es el que se encarga de compartir recursos en Windows. Si solo se comparten recursos entre máquinas Linux, el servicio nativo para ese fin es el servicio NFS que se verá en libro siguiente de esta misma unidad.

Características del servicio Samba

- ✓ Se utilizan los puertos 137, 138, 139 y 443. Puertos 137 y 138 con protocolo UDP y 139 y 443 con protocolo TCP.
- ✓ Los servicios en Linux, se configuran en un archivo. En el caso de Samba el **archivo de configuración es /etc/samba/smb.conf**
- ✓ Cuando se tenga el recurso compartido, desde un cliente Linux, podremos acceder al recurso con el comando mount y sistema de ficheros cifs; desde un cliente Windows, a través del explorador en Red o con la ruta UNC.

El servicio Samba se instala y configura en el ejercicio 3 de la tarea.

Paso 1. Instalación del servidor Samba. En máquina router.

Instalar SAMBA, con los 2 paquetes siguientes (se va a instalar la versión 4.19 que se encuentra en repositorios de Ubuntu 24.04):

```
root@SistemasUbuntu:~# apt-get install samba samba-common-bin
```

Comprobar si está activo SAMBA. Para ello, hay que saber que SAMBA está compuesto de 2 demonios: smbd y nmbd. Comprobamos con service que están corriendo. Se verá running en verde:

```
root@SistemasUbuntu:~# systemctl status smbd
```

..... (para salir pulsar q)

```
root@SistemasUbuntu:~# service nmbd status
```

Paso 2. Configuración del archivo /etc/samba/smb.conf

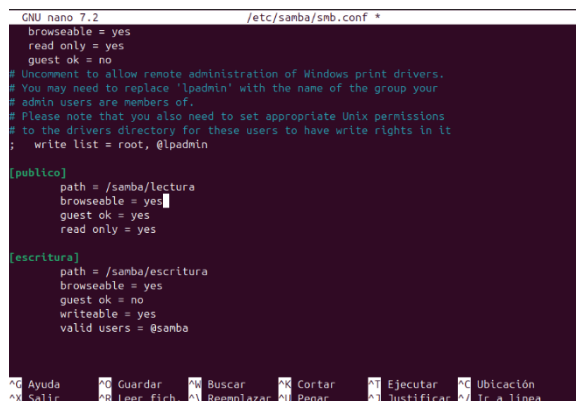
Editar el archivo **smb.conf** y donde aparece workgroup, rellenar el nombre del grupo de trabajo de tus máquinas windows de la unidad 9, por defecto como no lo hemos modificado será WORKGROUP. (Si apareciera la línea comentada habría que descomentarla)

workgroup = NOMBRE_GRUPO_TRABAJO

Para este ejemplo, se van a compartir 2 recursos. Una carpeta pública para todos los usuarios, con permisos de solo lectura y otra privada para algunos usuarios, con permisos de lectura y escritura. Para ello, **añadir al final de archivo smb.conf:**

[publico]
path = /samba/lectura
browseable = yes
guest ok = yes
read only = yes

[escritura]
path = /samba/escritura
browseable = yes
guest ok = no
writeable = yes
valid users = @samba



Significado de las distintas etiquetas:

- ✓ La etiqueta browseable = yes, sirve para que los usuarios conectados, puedan ver en el explorador el recurso, sino es un recurso secreto.
- ✓ En el recurso lectura se ha permitido el acceso a invitados (guest ok = yes), sin embargo, en el recurso escritura no (guest ok = no)
- ✓ En escritura, se ha permitido el acceso a los usuarios que pertenezcan al grupo samba de nuestro Linux (valid users = @samba). También se podría poner usuarios concretos, sin utilizar @. Por ejemplo para permitir acceso a juan y a los usuarios del grupo samba se especifica:
valid users = juan, @samba

Paso 3. Creación de usuario, grupos, carpetas y permisos en la máquina servidor.

Vamos a crear el grupo samba, e introducir en dicho grupo a los usuarios juan y juana. (juan es un usuario nuevo, y juana un usuario de los creados en la práctica de la unidad 9)

```
root@SistemasUbuntu:~# adduser juan
root@SistemasUbuntu:~# addgroup samba
root@SistemasUbuntu:~# adduser juan samba
root@SistemasUbuntu:~# adduser juana samba
```

#Creamos la carpeta samba y subcarpetas lectura y escritura. Cambiamos la propiedad de la carpeta al grupo samba, y sus permisos, de forma que puedan guardar cambios todos los usuarios del grupo y cualquiera pueda leer.

```
root@SistemasUbuntu:~# mkdir /samba
root@SistemasUbuntu:~# mkdir /samba/lectura
root@SistemasUbuntu:~# mkdir /samba/escritura
root@SistemasUbuntu:~# chgrp samba -R /samba
root@SistemasUbuntu:~# chmod 775 -R /samba
```

Estas tres líneas se pueden sustituir por una sola:
mkdir -p /samba/lectura /samba/escritura

Todos estos comandos son comandos de administración de Linux explicados en las unidades 5 y 6. Ahora, se utilizan comandos específicos de Samba, para añadir a los usuarios juan y juana como usuarios del servicio samba. Para ello:

```
root@SistemasUbuntu:~# smbpasswd -a juan
root@SistemasUbuntu:~# smbpasswd -a juana
```

Al ejecutar estos comando nos pedirá crear la contraseña samba de estos usuarios, esta contraseña no es necesario que sea la misma que la que tiene el usuario UNIX pero sí es recomendable que lo sea para no dar lugar a fallos de usuario al introducir las contraseñas.

Con el comando **pdbedit -L** podremos ver el listado de usuarios samba creados.

Una vez realizados cambios en los archivos de configuración de cualquier servicio, hay que reiniciarlo para que surtan efecto los cambios. En el caso de Samba, basta con reiniciar el servicio smbd:

```
root@SistemasUbuntu:~# systemctl restart smbd
```

2.1.- Conexión desde máquinas cliente.

Conexión desde máquinas clientes Windows

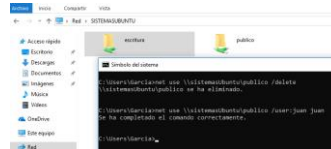
Iniciar máquina cliente1 de la unidad 9. Recuerda, que hemos configurado en archivo de samba, el nombre del grupo de trabajo. Conectar a la máquina a través del explorador con Red o con la ruta UNC \\SistemasUbuntu.

Se abrirá la ventana con los recursos compartidos lectura y escritura.

Al pulsar doble clic para realizar la conexión se solicita usuario, conectar con juan.

Si sale el mensaje de que no se tiene acceso, ejecutar los 2 comandos que se ven en la captura.

A continuación, se accede a ambas carpetas, con los permisos correctos.



Miguel Ángel García Lara (CC BY-NC-SA)

Conexión desde máquinas clientes Linux

Para realizar la conexión desde la máquina “clienteLinux” seguir los pasos siguientes:

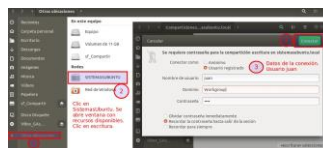
Instalar software para el cliente:

```
root@clienteLinux:~# apt install samba-common-bin
```

```
root@clienteLinux:~# apt install cifs-utils
```

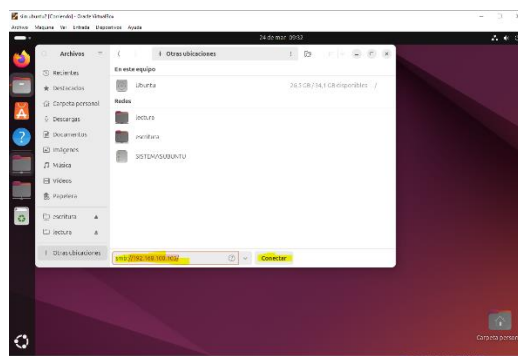
Realizar conexión gráfica

Desde el explorador de archivos Nautilus, si pulsamos en RED, ya se ve la máquina SistemasUbuntu. En la captura se realiza una conexión al recurso escritura con el usuario juan.



Miguel Ángel García Lara (CC BY-NC-SA)

En el caso de que no aparezca el icono de la máquina, podemos conectarnos a ella utilizando el protocolo Samba de la siguiente forma:



Ana Belén Ocón Pérez de Albéniz (CC BY-NC-SA)

Realizar conexión desde terminal. Comando mount:

```
root@clienteLinux:~# mkdir /mnt/escritura
```

#Se utiliza el comando mount, con sistema de ficheros cifs y usuario juan.

```
root@clienteLinux:~# mount -t cifs -o user=juan,pass=clave //192.168.100.103/escritura /mnt/escritura
```

#Donde pone clave ponderemos la contraseña del usuario samba juan

```
root@clienteLinux:~# mkdir /mnt/escritura/carpeta2
```

#Se ha creado carpeta2 en servidor Linux

¿Qué hacer para que el usuario no tenga que utilizar el comando mount cada vez que se conecte?

Se añade en /etc/fstab la línea de montaje automático (en la unidad 6)

```
root@clienteLinux:~# nano /etc/fstab
```

#Añadir al final del archivo la línea siguiente:

```
//192.168.100.103/escritura /mnt/escritura cifs rw,username=juan 0 0
```

#Al iniciar el equipo, antes de comenzar sesión en GNU-Linux se pregunta usuario de acceso a los recursos de Samba.

Autoevaluación

¿Cuál es falsa sobre Samba?

- ☐ Samba permite compartir recursos entre máquinas Windows y Linux
- ☐ El fichero de configuración es /etc/samba/smb.conf
- ☐ Utiliza 2 puertos
- ☐ Para montar un recurso se utiliza la opción cifs en el comando mount

Incorrecto

Incorrecto

Opción correcta

Incorrecto

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

3.- Servicio NFS.

Instalación y configuración del servidor NFS

El servicio NFS comparte recursos entre máquinas Linux, y es también compatible con máquinas Windows instalando la característica necesaria. El servicio NFS es más seguro que Samba.

Una vez instalado el servicio, se configurarán las carpetas a compartir en el **archivo de configuración /etc/exports**. A continuación, se reinicia el servicio.

Para la conexión de los clientes se utiliza el comando mount.

El servicio NFS se instala y configura en el ejercicio 4 de la tarea.

Paso 1. Instalar servidor NFS

```
root@SistemasUbuntu:~# apt install nfs-kernel-server
```

Paso 2. Configurar que recursos se comparten en el archivo /etc/exports

```
root@SistemasUbuntu:~# nano /etc/exports
```

#Añadir al final del fichero las 2 líneas siguientes:

```
/nfs/escritura 192.168.100.104(rw)
```

```
/nfs/lectura 192.168.100.0/24(ro)
```

#La primera línea comparte con escritura (read write) exclusivamente al equipo 192.168.100.104.

#La segunda línea comparte solo lectura (read only) a todos los equipos de nuestra red (observa que queda totalmente definida, pues se pone la dirección de red y los bits de la máscara).

Paso 3. Se crean las carpetas y se cambian los propietarios al usuario nobody y grupo nogroup

```
root@SistemasUbuntu:~# sudo mkdir /nfs
```

```
root@SistemasUbuntu:~# sudo mkdir /nfs/lectura
```

```
root@SistemasUbuntu:~# sudo mkdir /nfs/escritura
```

#Para que no haya problema de acceso, la carpeta tiene que pertenecer al usuario nobody y al grupo nogroup (usuario y grupo genéricos de Linux para servicios)

```
root@SistemasUbuntu:~# sudo chown -R nobody:nogroup /nfs
```

#Se crea un archivo en la carpeta lectura

```
root@SistemasUbuntu:~# echo hola > /nfs/lectura/saludo.txt
```

#Se cambian los permisos, de forma que puedan realizar todos los cambios en la carpeta el usuario y grupos propietarios:

```
root@SistemasUbuntu:~# chmod -R 777 /nfs
```

Se reinicia el servidor, de esa forma se lee el archivo /etc/exports y comprueba la existencia de los directorios compartidos.

```
root@SistemasUbuntu:~# systemctl restart nfs-kernel-server
```

3.1.- Cliente NFS.

Instalar cliente NFS en equipo clienteLinux y montar recurso con mount

Instalar cliente NFS

```
root@clienteLinux:~# apt install nfs-common
```

Crear las carpeta donde se van a montar los recursos

```
root@clienteLinux:~# mkdir /mnt/nfs
root@clienteLinux:~# mkdir /mnt/nfs/lectura
root@clienteLinux:~# mkdir /mnt/nfs/escritura
```

Montar el recurso de lectura. Comprobar que se tiene lectura y no escritura

```
root@clienteLinux:~# mount -t nfs 192.168.100.103:/nfs/lectura /mnt/nfs/lectura
```

Se comprueba que se puede leer el archivo saludo.txt creado en el servidor

```
root@clienteLinux:~# ls -l /mnt/nfs/lectura
```

```
total 4
```

```
-rw-r--r-- 1 root root 5 abr 23 11:56 saludo.txt
```

```
root@clienteLinux:~# cat /mnt/nfs/lectura/saludo.txt
```

```
hola
```

Se comprueba que no se puede escribir:

```
root@clienteLinux:~# echo soyCliente > /mnt/nfs/lectura/cliente.txt
```

```
bash: /mnt/nfs/lectura/cliente.txt: Sistema de archivos de solo lectura
```

Montar el recurso de escritura. Se comprueba que se puede escribir

```
root@clienteLinux:~# mount -t nfs 192.168.100.103:/nfs/escritura /mnt/nfs/escritura
```

```
root@clienteLinux:~# echo soyCliente > /mnt/nfs/escritura/cliente.txt
```

Archivo /etc/fstab líneas para no montar cada vez que se inicia el PC

Para no tener que utilizar el comando mount en cada sesión, se añaden las líneas correspondientes en el archivo /etc/fstab

```
root@clienteLinux:~# sudo nano /etc/fstab
```

#Añadir las 2 líneas siguientes:

```
192.168.100.103:/nfs/lectura    /mnt/nfs/lectura    nfs    ro,intr,x-gvfs-show    0    0
192.168.100.103:/nfs/escritura  /mnt/nfs/escritura  nfs    rw,intr,x-gvfs-show    0    0
```

Observaciones sobre las opciones:

- ✓ Se monta con sistema nfs, y las opciones son:
- ✓ ro/rw (read only/ read write según permiso en carpeta de servidor)
- ✓ intr: evita bloqueo en caso de error
- ✓ x-gvfs-show: que se monte gráficamente, además de terminal

Al iniciar el equipo, además de tener acceso en terminal, en el explorador de archivos Nautilus, habrá un acceso directo a las 2 carpetas.

4.- Servicio ssh.

Accesos remotos

El servicio acceso remoto consiste en acceder desde un equipo a otro de la red, y trabajar como si físicamente se estuviera en él. Los servicios más utilizados para acceder de forma remota a un sistema GNU/Linux son:

- ✓ Telnet. Permite acceder al sistema de forma remota por terminal, pero de una manera no segura.
- ✓ Open SSH. Permite acceder al sistema por terminal, pero de forma segura ya que se cifran las comunicaciones. Además, permite transferir archivos.
- ✓ VNC. Mientras que los servicios telnet y SSH permiten conectarse al servidor por medio de un terminal, el servidor VNC permite utilizar el servidor utilizando el escritorio instalado en el sistema: GNOME o KDE.

Parámetros de ssh e instalación

- ✓ El servicio ssh se puede instalar tanto en Windows como en Linux. Igualmente se puede acceder con clientes de ambos sistemas.
- ✓ El servicio ssh utiliza el puerto 22.
- ✓ Los usuarios de conexión, son los propios de la máquina Linux donde está instalado el servidor.
- ✓ El servicio ssh tiene dos ventajas muy importantes con respecto al servicio Telnet:
 - El servicio ssh es seguro a diferencia de telnet.
 - El servicio ssh además de acceso remoto permite la transferencia de ficheros con la utilidad scp.

En el ejercicio 5 de la tarea se instala y se realizan conexiones con el servicio ssh.

Para instalar el servidor ssh se ejecuta en la máquina servidor:

```
root@SistemasUbuntu:~# apt install openssh-server
```

#Comprobamos que ya se está ejecutando

```
root@SistemasUbuntu:~# systemctl status ssh
```

En el **archivo de configuración /etc/ssh/sshd_config** se configuran usuarios y/o equipos cliente que se les permite conectar.

En este caso, no le vamos a configurar. Por defecto se permite conectar desde toda la red y todos los usuarios del equipo.

Conexión desde equipos cliente

En el cliente no hay que instalar nada, simplemente conectarse con:

```
$ ssh user@IP
```

```
$ ssh user@Nombre_Equipo
```

Se está utilizando acceso remoto, de forma que físicamente se va a trabajar en la máquina clienteLinux, pero controlando la máquina servidor SistemasUbuntu. Por lo que tendremos que conectarnos con un usuario de la máquina servidor. Si nos conectamos con un usuario y password configurado en ambas máquinas, bastará para realizar la conexión con: **\$ ssh IP**

Ejemplo de conexión

```
ana@clienteLinux:~$ ssh 192.168.100.103
```

#conectar al servidor como ana

```
.....  
ana@192.168.100.103's password:
```

```
.....  
ana@SistemasUbuntu:~$ exit
```

#se observa que la propia shell dice que la máquina actual es SistemasUbuntu. Se puede ejecutar cualquier comando como si se estuviera físicamente en SistemasUbuntu.

#Al escribir exit, se sale de la sesión, y se vuelve al equipo cliente.

```
logout
```

```
Connection to 192.168.100.103 closed.
```

```
ana@clienteLinux:~$ ssh juan@192.168.100.103
```

#Se vuelve a realizar conexión al servidor, ahora como juan. El usuario juan existe en la máquina servidor, pero

no en la máquina cliente.

juan@192.168.100.103's password:

.....

#Se pregunta quien está conectado (who). Devuelve el usuario miguel en la propia máquina SistemasUbuntu y el usuario juan, desde la red, desde la máquina 192.168.100.104

juan@SistemasUbuntu:~\$ **who**

estudiante seat0 2025-03-20 20:21 (login screen)

estudiante tty2 2025-03-20 20:24 (tty2)

juan pts/1 2025-03-20 22:52 (192.168.100.104)

juan@SistemasUbuntu:~\$ **exit** #Salir de ssh

logout

Connection to 192.168.100.103 closed.

Transferir archivos con comando scp.

El servicio ssh, incorpora el comando scp para copiar ficheros de la máquina servidor a la máquina cliente, o viceversa. Dicho de otra forma, el servicio ssh incorpora con el **comando scp** un servicio ftp que además es seguro y se conoce como **sftp**.

Observación

El comando scp se utiliza sin necesidad de realizar una conexión previa al servidor.

Sintaxis

scp [-r] [origen] [destino]

dónde, origen o destino lo forman: [usuario@maquina:archivo]

La opción -r es de recursiva para enviar o recibir una carpeta completa.

Ejemplo para enviar:

\$ scp practica.odt juan@192.168.100.103:/home/juan/practica.odt

Se envía el archivo practica.odt al servidor ssh 192.168.100.103

Ejemplo para recibir

\$ scp [juan@192.168.100.103:/home/juan/practica.odt](#) nombre.odt

Se recibe el archivo practica.odt desde el servidor ssh 192.168.100.103

5.- Servicio Web: Apache.

Instalación y configuración del servicio web Apache

En la unidad 9 se instaló Internet Information Services de Windows con su servidor web y servidor ftp. En este apartado se va a explicar la configuración del servidor web Apache. Apache se puede instalar tanto en Windows como en Linux. En la tarea no se ha incluido ningún ejercicio relacionado con Apache, no por su poca importancia, sino por haber instalado ya un servicio web en Windows.

Hace 20 años se creó la "Apache Software Foundation". Esta fundación la forman desarrolladores de software donde cada uno realiza sus propios proyectos de código abierto.

Página de Apache: <http://www.apache.org/>

Página de la fundación de software de Apache: https://es.wikipedia.org/wiki/Apache_Software_Foundation

Para instalar Apache en GNU-Linux: **# apt-get install apache2**

Una vez instalado, ya está funcionando Apache. Para comprobarlo, ejecutar: **#systemctl status apache2**

Ya se puede navegar y ver la página por defecto del servidor web Apache:

- ✓ Desde el equipo servidor, escribir en el navegador: <http://localhost>
- ✓ Desde otra máquina, escribir en el navegador: http://IP_servidor

Apache devuelve la página por defecto index.html alojada en el directorio /var/www/html



Miguel Ángel García Lara (CC BY-NC-SA)

Configuración del servidor Apache

La configuración de apache se almacena en el **directorio de configuración /etc/apache2**.

A continuación se ven las **opciones de configuración** más importantes:

- ✓ El archivo **/etc/apache2/apache2.conf** tiene la configuración global del servidor.
- ✓ El archivo **/etc/apache2/ports.conf**. Permite **establecer los puertos de escucha** de Apache, por defecto **puerto 80 para http y puerto 443 para https**.
Listen *:80
Listen *:443
El * significa que puede realizar la petición cualquier ordenador.
- ✓ En el mismo servidor se pueden tener varios sitios web, para ello, en la carpeta **/etc/apache2/sites-available** se guarda la configuración de cada uno en un archivo. Por defecto se encuentran los sitios default y default-ssl. Cada sitio (o archivo) tiene la siguiente estructura:
<VirtualHost *:80>
ServerAdmin servermaster@localhost
ServerName www.miempresa.com
DocumentRoot /var/www/html
DirectoryIndex index.html default.html
</VirtualHost>

Dónde el significado de estas 4 directivas es el siguiente:

- ✓ ServerAdmin es el correo electrónico del administrador del sitio web.
- ✓ ServerName es el nombre FQDN del sitio web. Para el dominio por defecto (archivo default) no se indica ningún nombre, por lo que la línea se deja comentada. Pero para otros dominios (por ejemplo, www.miempresa.com) sí se debe establecer.
- ✓ DocumentRoot. Indica la ubicación donde se encuentra las páginas web del sitio.

- ✔ DirectoryIndex. Indica el nombre de los ficheros que envía por defecto el servidor web. En el ejemplo, significa, que cuando en el navegador se escriba `http://IP_servidor` se devuelve la página `/var/www/html/index.html`

Ejemplo de creación de un nuevo sitio web

En un servidor web Apache se pueden alojar distintas páginas web o sitios web. Por defecto el servidor web publica el directorio `/var/www/html` para todos los dominios pero es posible personalizar de forma independiente cada dominio.

- ✔ Para añadir el dominio `www.miempresa.com` que se aloje en la carpeta `/portales/miempresa` hay que **crear el fichero `/etc/apache2/sites-available/miempresa.com`** con el siguiente contenido:

```
<virtualhost *:80>
ServerName www.miempresa.com
DocumentRoot /portales/miempresa
DirectoryIndex index.html index.htm
</virtualhost>
```
- ✔ Activar el sitio
`# a2ensite miempresa.com`
- ✔ Reiniciar el servidor web
`# service apache2 restart`

Página de inicio del servidor https

Con el auge de los negocios en Internet se ha popularizado el uso de comunicaciones cifradas entre los clientes y el servidor Web, siendo la tecnología de encriptación más utilizada el Security Socket Layer (SSL).

Por defecto viene instalado en Apache, por lo que para utilizar una página segura bajo https en Apache sólo hay que:

- ✔ Activar el módulo ssl
- ✔ Activar el sitio default-ssl
- ✔ Reiniciar el servidor web

Una vez reiniciado el servidor, se accede desde el navegador con la dirección `https://IP_Servidor`. Se puede generar certificado de seguridad utilizando el comando `open-ssl`.

5.1.- Apache con PHP.

Páginas web dinámicas lado servidor con PHP

PHP es un lenguaje de programación utilizado para crear páginas web dinámicas del lado servidor. En las páginas hay scripts de php para distintos fines. Como ejemplo de utilización, suponer que se solicita usuario y contraseña al usuario del navegador web, el código php accede a la base de datos guardada en el servidor, comprobando si el usuario tiene acceso y devolviendo los datos requeridos.

Para instalar php en nuestro servidor web Apache ejecutar:
apt install php

En Ubuntu 24.04 se instala la versión 8.3 de php.

```
estudiante@sistemasubuntu:~$ php --version
PHP 8.3.6 (cli) (built: Dec 2 2024 12:36:18) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
estudiante@sistemasubuntu:~$
```

Ana Belén Ocón Pérez de Albéniz (CC BY-NC-SA)

Para comprobar que PHP se ha instalado con éxito se puede crear un fichero php y ubicarlo en el directorio raíz del servidor web.

Ejemplo: Crear script de inicio info.php

- ✓ Editar el fichero /var/www/html/info.php.
nano /var/www/html/info.php # Escribir el contenido siguiente
<?php
echo "La información de mi servidor:<p />";
phpinfo();
?> **#Guardar archivo**
- ✓ Ahora, en el navegador web se escribe en la barra de direcciones <http://localhost/info.php>. Esta página devuelve la información del servidor utilizado (instrucción phpinfo).

Observar que ahora hemos tecleado el nombre del archivo, pues por defecto devolvería index.html (al ser el indicado en la directiva DirectoryIndex)



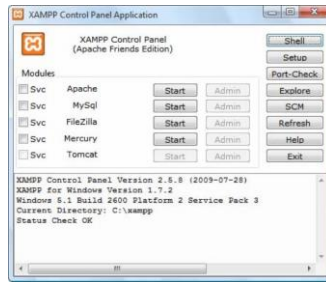
Miguel Ángel García Lara (CC BY-NC-SA)

XAMPP

XAMPP es un paquete que aglutina varios servicios y programas: el servidor web Apache, el gestor de bases de datos MySQL y los intérpretes de lenguaje de scripts PHP y Perl. De esa forma, los programadores de páginas dinámicas utilizan este software, como una opción simple para tener todos los servicios instalados en un único proceso y con una configuración sencilla.

XAMPP se puede instalar tanto en GNU-Linux como en Windows.

Una vez instalado, hay un panel de control para iniciar o detener cada servicio (ver imagen)



Materiales FP a Distancia del MEC (Dominio público)

Una vez instalado e iniciado Apache, ya se puede visualizar en el navegador la página de inicio de XAMPP escribiendo en el navegador `http://localhost`.

Esta página de inicio está guardada en el directorio de publicación del servidor web Apache por defecto, que en el caso de Windows es `C:\xampp\htdocs`. Ya se puede guardar en esa carpeta cualquier página web estática (html) o página dinámica (php)

Autoevaluación

En Apache ¿en qué directorio se alojan por defecto las páginas web?

- ☐ `/var/www/html`
- ☐ `/www/apache2`
- ☐ `/var/html`
- ☐ `/etc/www/html`

Opción correcta

Incorrecto

Incorrecto

Incorrecto

Solución

1. Opción correcta
2. Incorrecto
3. Incorrecto
4. Incorrecto

6.- Servicio FTP: vsftpd

Configuración

El servicio ftp utiliza los puertos 20 y 21.

El servicio ftp más utilizado en Linux se llama **vsftpd** (Very Secure FTP) es un servidor FTP muy pequeño y seguro.

Para instalarlo: **# apt install vsftpd**

Una vez instalado y comprobado que está funcionando, ya se puede conectar el navegador. Las conexiones se harán con los mismos clientes ftp que los vistos en la unidad 9: terminal, filezilla, navegadores web.

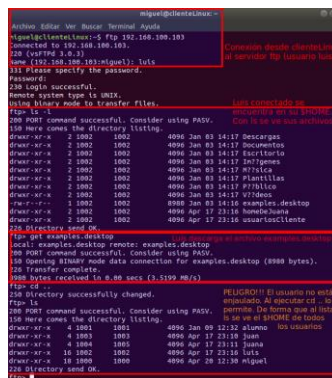
El archivo de configuración se llama **/etc/vsftpd.conf**

En este archivo se configura los usuarios permitidos, si se permite usuario anónimo o no, como la carpeta de descarga del servidor.

Por defecto, no se permite el acceso al usuario anónimo. Los usuarios permitidos son los propios del sistema, y su directorio de conexión es su \$HOME. Es decir, si se conecta el usuario luis al servidor ftp, el directorio donde se conecta es /home/luis.

Además, se configura si los usuarios se quedan enjaulados en su directorio, este concepto es importante. Supongamos que el directorio de conexión ftp de un usuario es /home/luis es importante asegurarse que no va a poder salir al directorio padre con el comando **cd ..**

De hecho, por defecto los usuarios no están enjaulados. Ver captura.



Miguel Ángel García Lara (CC BY-NC-SA)

Para saber más

Para ver más información del fichero de configuración, ver vínculo:

[Cómo instalar y configurar VSFTPD Ubuntu 18](#)

Anexo.- Licencias de recursos.

Licencias de recursos utilizados en la Unidad de Trabajo.

Recurso (1)	Datos del recurso (1)	Recurso (2)	Datos del recurso (2)
	Autoría: rgtaylor_csc. Licencia: GPL. Procedencia: http://openclipart.org/detail/17668/net-computer-by-rgtaylor_csc		Autoría: Ubuntu. Licencia: GNU/GPL. Procedencia: Captura de pantalla del terminal ejecutando ifconfig, propiedad de Ubuntu.
	Autoría: Ubuntu. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la herramienta de configuración de red de Xwindows, propiedad de Ubuntu.		Autoría: Ubuntu. Licencia: Uso Educativo no comercial. Procedencia: Captura de pantalla de la herramienta de configuración de red inalámbrica de Xwindows, propiedad de Ubuntu.
	Autoría: Ubuntu. Licencia: GNU/GPL. Procedencia: Captura de pantalla del escritorio de Ubuntu, propiedad de Ubuntu.		Autoría: Andrew Fitzsimon / Anonymous. Licencia: GPL. Procedencia: http://openclipart.org/detail/25528/text-page-icon-by-anonymous-25528
	Autoría: Ubuntu. Licencia: GNU/GPL. Procedencia: Captura de pantalla del terminal mostrando el comando ping, propiedad de Ubuntu.		Autoría: Ubuntu. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la herramienta de red de Xwindows, propiedad de Ubuntu.
	Autoría: HASH(0x89c79d4) / Anonymous. Licencia: GPL. Procedencia: http://openclipart.org/detail/24075/firewall-by-anonymous-24075		Autoría: Andrew Fitzsimon / Anonymous. Licencia: GPL. Procedencia: http://openclipart.org/detail/25428/network-by-anonymous-25428
	Autoría: isc.org. Licencia: GPL. Procedencia: Captura de pantalla de www.isc.org .		Autoría: www.samba.org . Licencia: GPL. Procedencia: Captura de pantalla www.samba.org .
	Autoría: warszawianka. Licencia: GPL. Procedencia: http://openclipart.org/detail/35347/tango-system-users-by-warszawianka		Autoría: lyte. Licencia: GPL. Procedencia: http://openclipart.org/detail/163717/file-server-by-lyte
	Autoría: Ubuntu. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la herramienta Imprimiendo propiedad de Ubuntu.		Autoría: Ubuntu. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la herramienta Imprimiendo propiedad de Ubuntu.
	Autoría: swat. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la herramienta swat propiedad de SAMBA.		Autoría: webmin. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la herramienta webmin, propiedad de www.webmin.org .

			
	<p>Autoría: Ubuntu. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la herramienta Configuración del servidor de Samba propiedad de Ubuntu.</p>		<p>Autoría: Anonymous. Licencia: GPL. Procedencia: http://openclipart.org/detail/109591/another-folder-icon-01-by-anonymous</p>
	<p>Autoría: Blog del sensei. Licencia: GPL. Procedencia: Montaje sobre: http://josejuanlt.ipower.com/blogsensei/?p=763</p>		<p>Autoría: OpenSSH. Licencia: GNU/GPL. Procedencia: www.openssh.org.</p>
	<p>Autoría: Andrew Fitzsimon / Anonymous. Licencia: GPL. Procedencia: http://openclipart.org/detail/25528/text-page-icon-by-anonymous-25528</p>		<p>Autoría: Putty. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la aplicación Putty, propiedad de Putty.</p>
	<p>Autoría: tightVNC Software. Licencia: GNU/GPL. Procedencia: http://www.tightvnc.com/</p>		<p>Autoría: Vinagre. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la aplicación VINGRE, propiedad de Vinagre.</p>
	<p>Autoría: tightVNC. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la aplicación tightVNC, propiedad de tightVNC.</p>		<p>Autoría: Ubuntu y tightVNC. Licencia: GNU/GPL. Procedencia: Captura de pantalla de la aplicación tightVNC, propiedad de tightVNC.</p>
	<p>Autoría: Firefox y Apache. Licencia: GNU/GPL. Procedencia: Captura de pantalla del navegador Firefox, propiedad de firefox.</p>		<p>Autoría: Andrew Fitzsimon / Anonymous. Licencia: GPL. Procedencia: http://openclipart.org/detail/25528/text-page-icon-by-anonymous-25528</p>
	<p>Autoría: Firefox y PHP.org. Licencia: GNU/GPL. Procedencia: Captura de pantalla del navegador Firefox, propiedad de firefox .</p>		<p>Autoría: Firefox. Licencia: GNU/GPL. Procedencia: Captura de pantalla del navegador Firefox, propiedad de firefox.</p>
	<p>Autoría: vsftpd. Licencia: GNU/GPL. Procedencia: vsftpd.beasts.org/.</p>		