



Ciclo Formativo: Desarrollo de Aplicaciones Web

DESPLIEGUE DE APLICACIONES WEB

TEMA 2 – Configuración de una red local

Utilizando Ubuntu Desktop 16.04 y Ubuntu Server 16.04, sobre máquinas virtuales

Marcos Alcañiz

Contenido

1.	Diseño conceptual de la red	2
2.	Configuración de las máquinas virtuales en VirtualBox.....	4
3.	Instalación de los Sistemas Operativos	5
4.	Direccionamiento IP	6
4.1.	Configuración IP del Servidor	7
4.2.	Configuración IP del <i>Cliente 1</i>	9
4.3.	Configuración de la conexión externa.....	9
4.3.1.	Configuración DNS.....	10
4.3.2.	Configuración del protocolo NAT	11
5.	Comprobaciones	15

TEMA 2

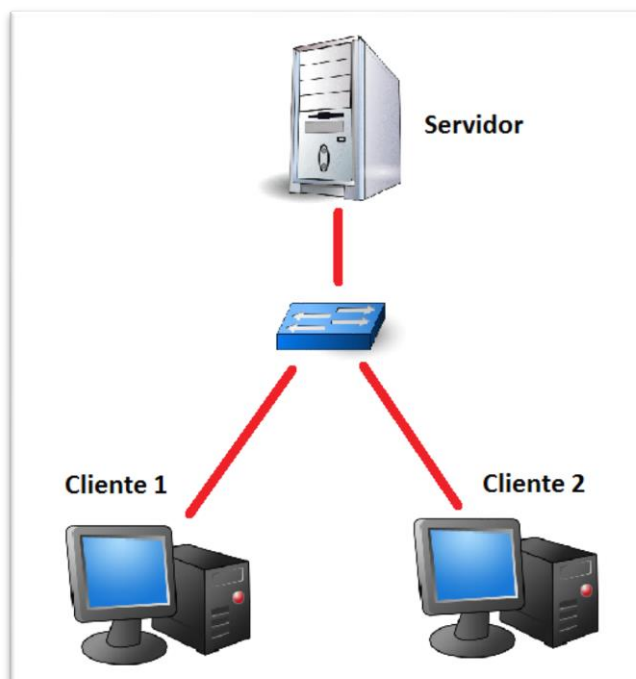
Configuración de una red local

En este tema introductorio aprenderemos a configurar una red LAN compuesta por un servidor que hará de router, y dos máquinas clientes.

La realizaremos sobre tres máquinas virtuales usando VirtualBox, de manera que nos sirva a lo largo del curso para instalar y probar los diferentes servicios y aplicaciones que veremos en este módulo.

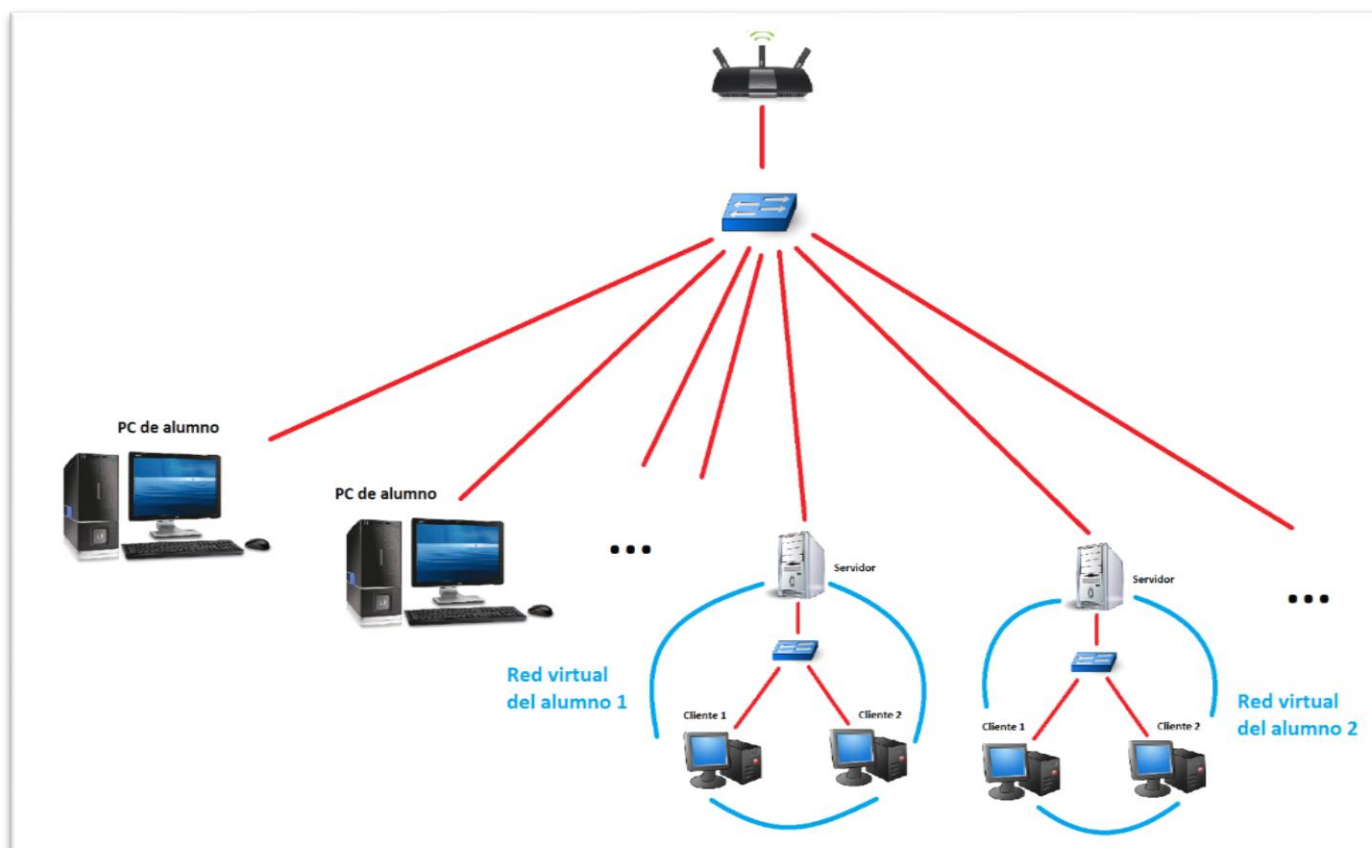
1. Diseño conceptual de la red

La red LAN que pretendemos configurar presenta la siguiente morfología:



Estas tres máquinas serán realmente máquinas virtuales. La idea es que cada alumno tenga montada una red interna de este tipo, pero que cada servidor sí esté conectado a la red del aula del instituto, de manera que en un futuro puedan dar servicios al resto de compañeros de clase.

Por tanto, el diseño de la red desde la perspectiva global del aula (o de nuestra casa) será el siguiente:

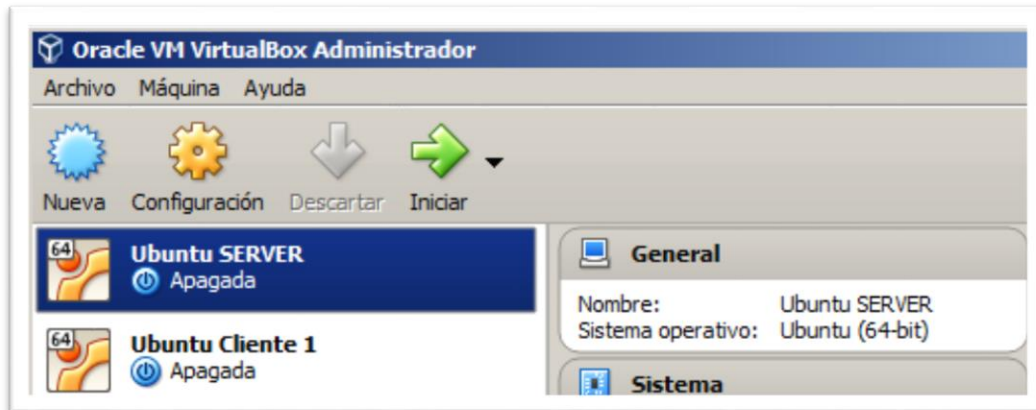


De esta manera, cada alumno deberá configurar tres máquinas: un servidor y dos clientes. Utilizaremos el sistema operativo Ubuntu, tanto su versión de escritorio para los clientes como la versión server para el servidor.

Puntualizar que los servidores deberán disponer de dos tarjetas de red: una para dar servicio a la red virtual interna, y otra para conectar esa red virtual a la red del aula.

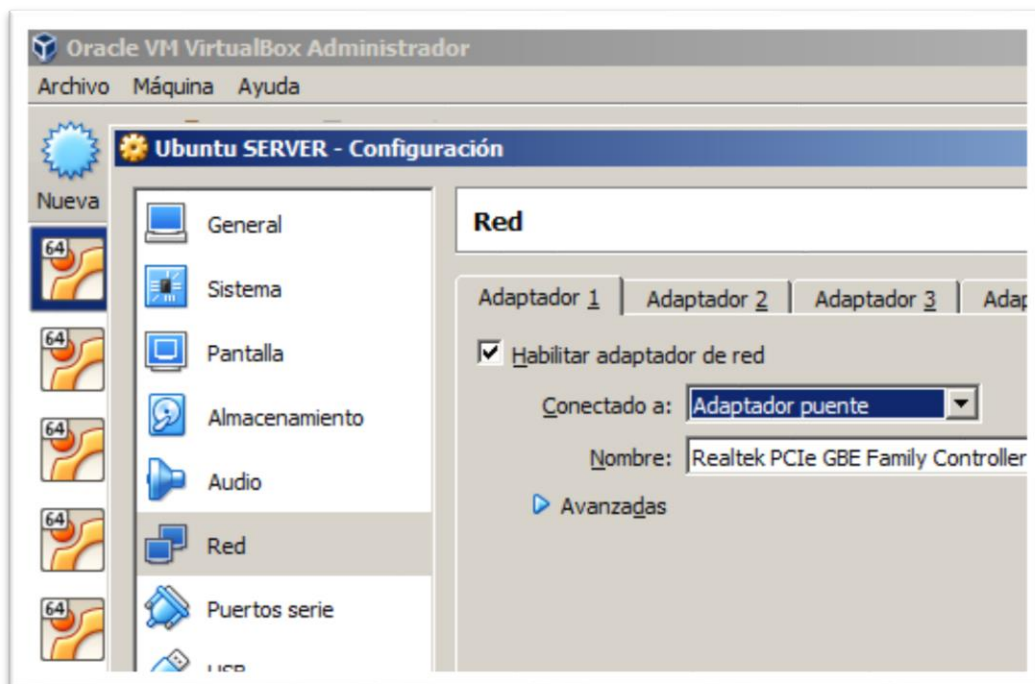
2. Configuración de las máquinas virtuales en VirtualBox

Crearemos la máquina virtual del *Servidor*, y la máquina virtual del *Cliente 1* (más adelante, cuando tengamos esta última configurada, la clonaremos para disponer del *Cliente 2*). En principio, con unos 10GB de memoria de disco duro deberíamos tener almacenamiento de sobra.



En la máquina *Servidor* deberemos habilitar dos adaptadores de red en *Configuración/Red*.

El primero de ellos lo configuraremos como “*Adaptador puente*”, y nos servirá para conectar la red virtual a la red real externa (ya sea del aula o de nuestra casa).



El segundo adaptador dará servicio a la red virtual interna (donde estarán conectados los dos clientes). Por lo tanto, lo configuraremos como “Red interna” y nos inventaremos un nombre, por ejemplo “red virtual DAW”.



3. Instalación de los Sistemas Operativos

Llegados a este punto, debemos descargar las ISO de los sistemas operativos que deseemos instalar en las máquinas de nuestra red.

En este caso, descargaremos las imágenes de *Ubuntu Desktop 16.04* para los clientes, y *Ubuntu Server 16.04* para la máquina del servidor. Por supuesto, podemos instalar los SO's que prefiramos.

Es posible que durante la instalación en el Servidor se nos pregunte cuál es la interfaz de red con la que conectar a internet. Si se diera el caso, seleccionaremos el adaptador de red que nos conecte con la red real externa (normalmente “*enp0s3*” en VirtualBox, o “*eth0*” en máquinas reales).

4. Direccionamiento IP

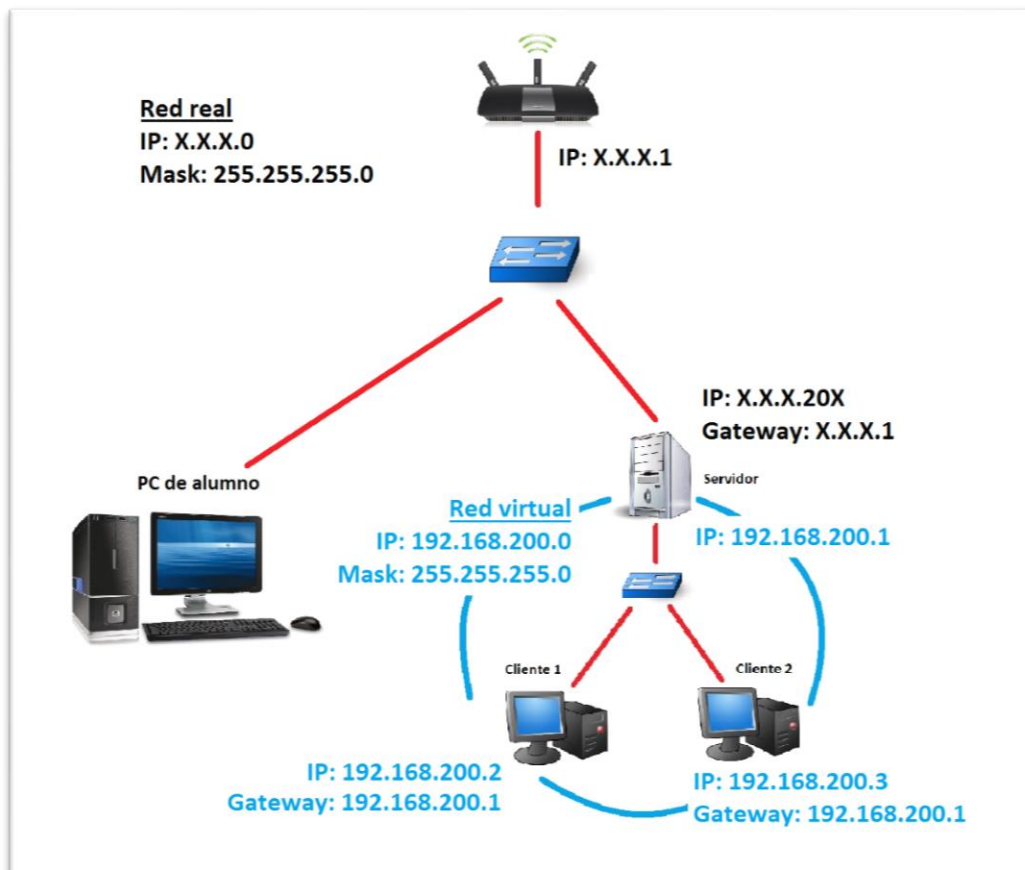
Tendremos que facilitar configuraciones IP válidas a todos los dispositivos de nuestra red.

¡IMPORTANTE! La red real externa (ya sea el aula o nuestra casa) tendrá diferentes direcciones IP dependiendo de dónde estemos y de cada situación. Por lo tanto nos referiremos a esas direcciones como **X.X.X.X/XX**, teniendo que sustituirlas por las direcciones IP reales que cada uno tengamos en nuestra casa o nuestra aula.

Además, como en una misma aula deben convivir varios servidores virtuales (uno por alumno), tendremos que ponernos de acuerdo para que ningún servidor repita dirección IP con ningún otro. Tomamos la decisión de asignar direcciones a partir de X.X.X.200, siendo éste el primer servidor, seguido del X.X.X.201, X.X.X.202... y así sucesivamente. Por tanto, nos referiremos a estas direcciones como **X.X.X.20X**.

Supondremos una configuración IP típica en la red real externa, y a nuestra red virtual le asignaremos la IP **192.168.200.0/24**. Esta dirección es arbitraria; podríamos elegir cualquiera otra red privada válida... aunque obviamente tendríamos entonces que tener en cuenta este cambio en todos los pasos siguientes.

Dicho esto, la propuesta de asignación que utilizaremos será la siguiente:



4.1. Configuración IP del Servidor

Para configurar la red en el *Servidor* podemos utilizar el comando **ifconfig**, o directamente el archivo de configuración **interfaces** situado en **/etc/network**. Nosotros nos decantaremos por este último.

Primeramente deberemos conocer el nombre de las dos interfaces de red de las que dispone el *Servidor*. Podemos visualizarlas con el comando:

```
~$ ifconfig -a
```

La opción “**-a**” nos muestra todas las interfaces, incluidas las desactivadas.

Los nombres de las interfaces normalmente (¡no siempre!) serán “*eth0*” y “*eth1*” en caso de máquinas reales, o “*enp0s3*” y “*enp0s8*” en máquinas virtuales de *VirtualBox*.

Tal y como hemos comentado anteriormente, la primera interfaz será la que conecte con la red real exterior, y la segunda interfaz conectará con nuestra red virtual interna.

NOTA: A la interfaz externa podríamos darle una configuración IP dinámica (es decir, que sea el router exterior el que proporcione la configuración). Esto sería muy cómodo ya que, no solamente nos evitaríamos tener que configurar esa interfaz manualmente, sino que además podríamos mover nuestra red virtual del aula de clase a nuestra casa sin tener que modificar absolutamente nada (ya que la configuración externa del Servidor se adaptaría automáticamente).

Sin embargo, esto tiene un inconveniente muy grande para un servidor: al tener una dirección IP cambiante, nos tocaría consultar su dirección IP cada vez que queramos acceder o incluso configurar los servicios que tenga alojados. Por esta razón, nosotros optaremos por configurarle manualmente la dirección **X.X.X.20X** que ya hemos comentado antes.

Deberemos acceder por tanto al archivo `/etc/network/interfaces` y editarlo añadiendo la configuración acordada:

```
GNU nano 2.5.3      Archivo: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# Interfaz externa
auto enp0s3
iface enp0s3 inet static
address X.X.X.20X
netmask 255.255.255.0
gateway X.X.X.1

# Interfaz interna
auto enp0s8
iface enp0s8 inet static
address 192.168.200.1
netmask 255.255.255.0
```

Para que la configuración sea efectiva, tendremos que reiniciar las interfaces (o simplemente habilitarlas, si aún estaban inhabilitadas). Podemos reiniciar toda la red:

```
~$ sudo /etc/init.d/networking restart
```

O reiniciar las interfaces individualmente:

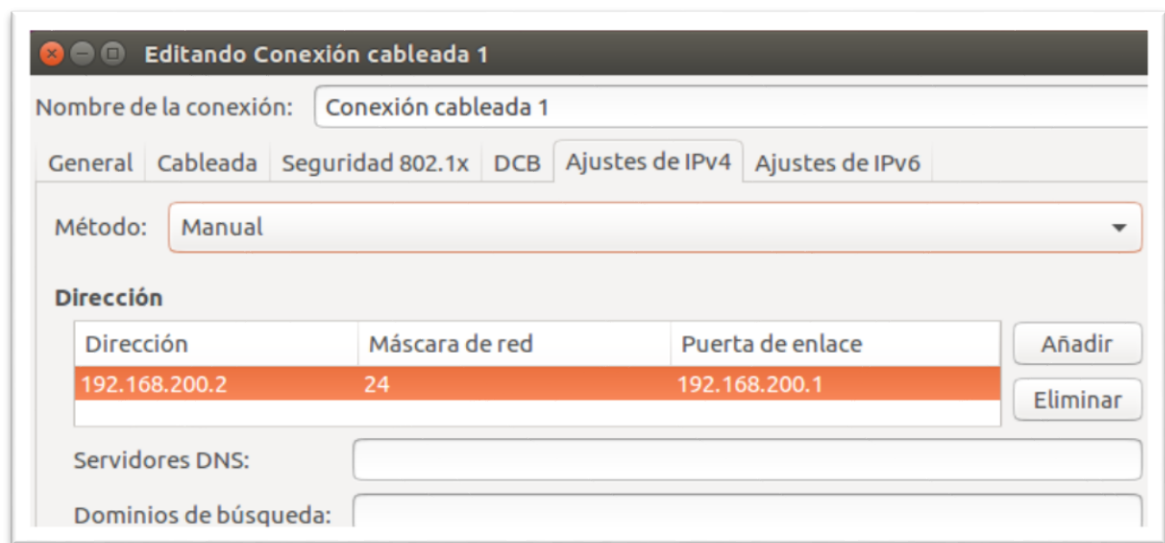
```
~$ sudo ifconfig enp0s3 down
~$ sudo ifconfig enp0s3 up
```

```
~$ sudo ifconfig enp0s8 down
~$ sudo ifconfig enp0s8 up
```

4.2. Configuración IP del Cliente 1

El *Cliente 1* utiliza *Ubuntu Desktop*, por lo que la configuración IP es ligeramente diferente: no se debe utilizar los archivos de configuración, sino que debemos utilizar la aplicación gráfica que *Ubuntu* nos proporciona (*utilizar los archivos de configuración genera conflictos con dicha aplicación, al menos en la versión Ubuntu 16.04 que utilizamos en el ejemplo!*).

Por ello, debemos irnos al icono de red situado en la esquina superior derecha, editar la única conexión que debería aparecernos, e introducir la configuración acordada en la pestaña “Ajustes de IPv4”:



También deberemos reconectar la red para que la configuración sea efectiva.

Llegados a este momento nuestra red LAN debería de ser funcional: tanto el *Servidor* como el *Cliente 1* son capaces de enviarse **pings** con éxito.

4.3. Configuración de la conexión externa

Hemos conseguido configurar el funcionamiento interno de nuestra red LAN virtual. Ha llegado el momento de proporcionarle conectividad con el exterior, de manera que nuestros equipos puedan tener conexión a internet.

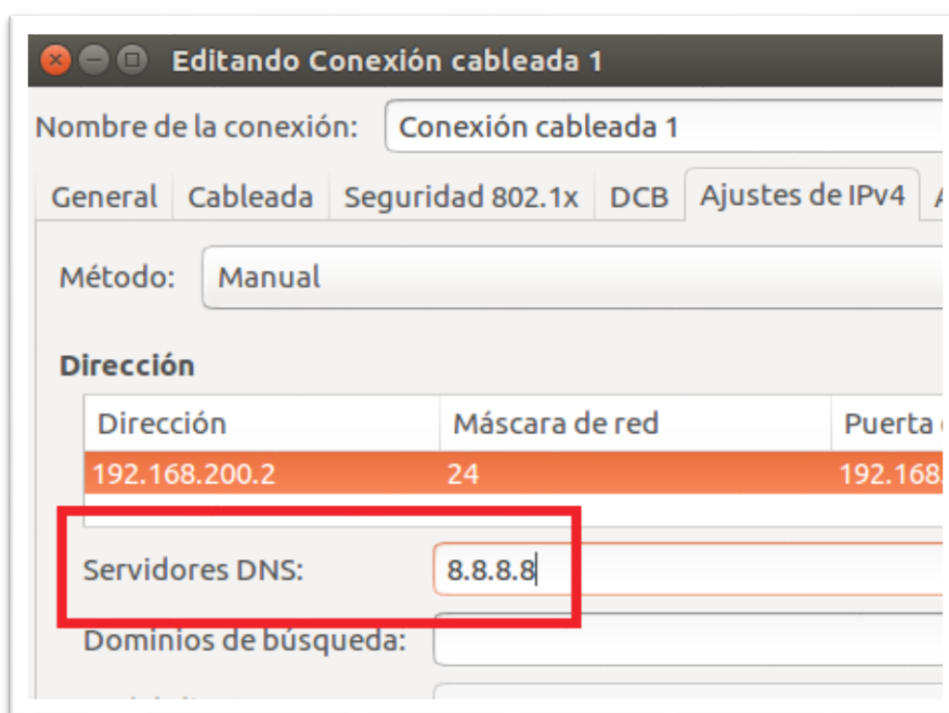
NOTA: Por inercia, hemos configurado anteriormente la interfaz externa de nuestro Servidor proporcionándole una puerta de enlace X.X.X.1. Técnicamente esa puerta de enlace no era necesaria, ya que no afecta para nada a la red virtual interna... pero era obvio que tarde o temprano íbamos a necesitarla si queremos darle conectividad externa a nuestra red.

4.3.1. Configuración DNS

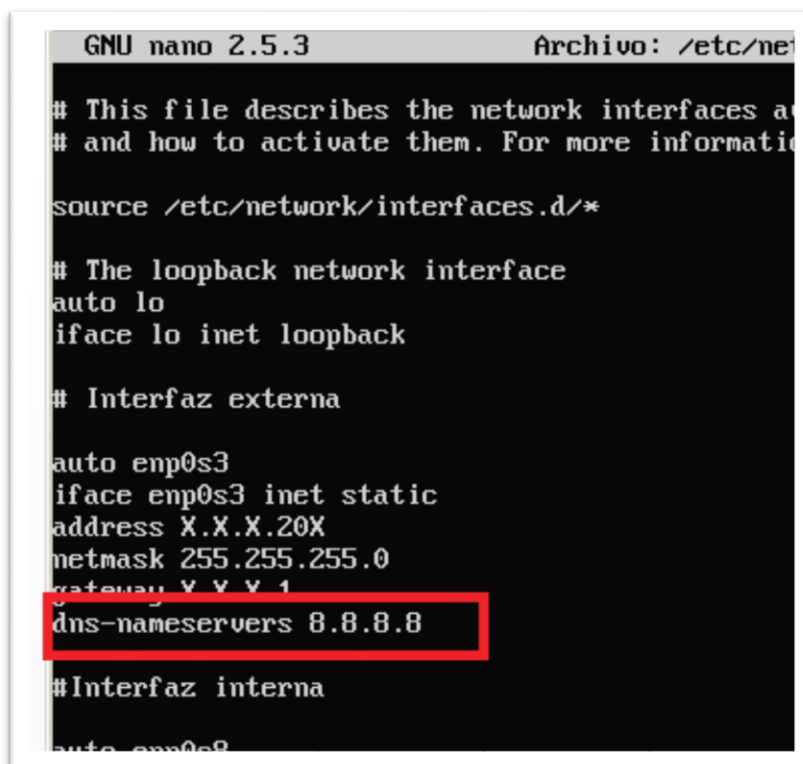
Por una parte, tendremos que configurar las direcciones DNS de todos los dispositivos. El servidor DNS que les indiquemos será el que se ocupe de traducir *URL*'s en direcciones IP, y viceversa.

Podemos instalar nuestro propio servidor DNS, o bien utilizar el DNS que nos ofrezca nuestro proveedor, o bien utilizar un servidor DNS público. Nos decantaremos por éste último, eligiendo el servidor público y gratuito de *Google* con dirección **8.8.8.8**.

En el *Cliente 1*, deberemos acceder de nuevo a la edición de nuestra conexión de red e introducir el DNS en la pestaña "*Ajustes de IPv4*":



Para configurar el DNS en la máquina *Servidor*, tendremos que editar otra vez el archivo */etc/network/interfaces* y añadir el servidor DNS a la interfaz externa:



```
GNU nano 2.5.3 Archivo: /etc/net
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see the man pages
of the /etc/network/interfaces file.

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# Interfaz externa
auto enp0s3
iface enp0s3 inet static
address X.X.X.20X
netmask 255.255.255.0
gateway X.X.X.1
dns-nameservers 8.8.8.8

# Interfaz interna
auto enp0s8
iface enp0s8 inet static
```

¡IMPORTANTE! En versiones anteriores de Ubuntu, y en muchas otras distribuciones Linux, los servidores DNS deben indicarse en el archivo */etc/resolv.conf*.

Aún a día de hoy, muchísimos manuales y tutoriales vigentes indican esta manera.

4.3.2. Configuración del protocolo NAT

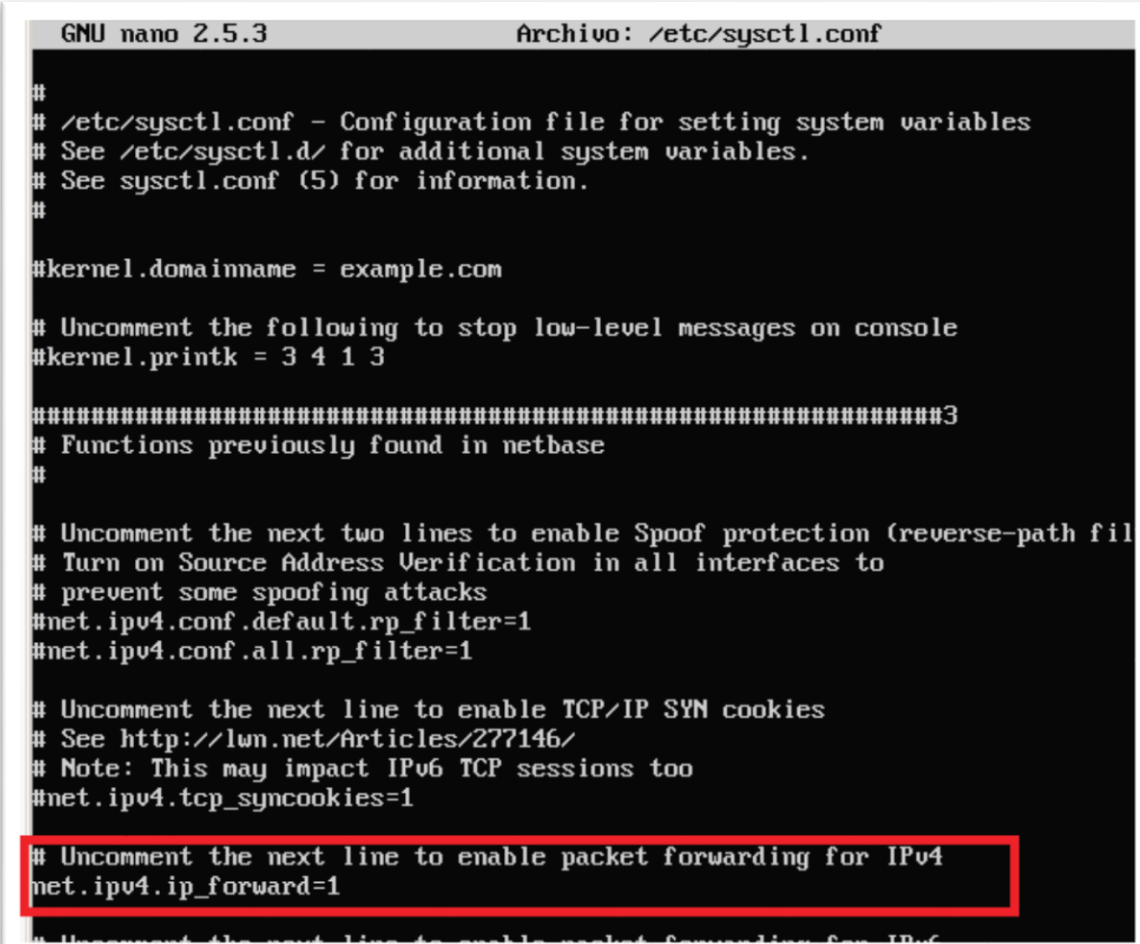
Con lo configurado hasta ahora, en teoría los paquetes con destino a *internet* deberían ser capaces de salir de nuestra red virtual interna (saltando de *puerta de enlace* en *puerta de enlace*) hasta llegar a su destino.

Sin embargo, es altamente probable que el destino se encuentre tan *lejos* de nuestra red, que le sea imposible contestarnos directamente. Para que la respuesta a ese paquete pueda volver de nuevo al origen, cada dispositivo ha tenido que ir “*mintiendo*” al siguiente haciéndole creer que el origen de la comunicación era él mismo (de esta manera, cada dispositivo tiene *muy cerca* al dispositivo al que tiene que contestar).

El protocolo responsable de ese procedimiento se denomina NAT (*Network Address Translation*), y debe estar activado en todos los dispositivos intermedios para que la completa conexión con el exterior sea posible (*técnicamente, con tablas de enrutamiento podría solucionarse también este problema... pero esa es otra historia*).

Para poner en marcha este protocolo NAT en nuestro *Servidor*, primero deberemos indicarle que permita el enrutamiento de paquetes a través de él. Esto se consigue activando la característica **IP Forwarding**.

Para ello, existen diferentes métodos... pero si queremos que esta opción se quede activa tras el apagado del *Servidor*, tendremos que editar el archivo **/etc/sysctl.conf** y asegurarnos de que el **bit forward** para IPv4 tenga el valor **1**. Para ello buscamos la línea en cuestión, y la descomentamos (o simplemente la añadimos):



```
GNU nano 2.5.3          Archivo: /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com
#
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#
#####3
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path fil
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
#
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
# Uncomment the next line to enable packet forwarding for IPv6
```

Ahora crearemos una regla de enrutamiento para que nuestro *Servidor* “mienta” al dispositivo siguiente sustituyendo la dirección de origen de los paquetes por la suya propia, tal y como hemos comentado.

Para ello, utilizaremos la aplicación **iptables**:

```
~$ sudo iptables -t nat -A POSTROUTING -s 192.168.200.0/24 -o enp0s3 -j SNAT --to X.X.X.20X
```

Donde:

-t nat: indica que queremos modificar la tabla NAT.

-A POSTROUTING: indica que queremos añadir (*ADD*) una regla que se aplique después de que el destino del paquete se haya decidido (*POSTROUTING*).

-s 192.168.200.0/24: indica que la regla se aplicará a todos los paquetes que provengan (*source*) de la red indicada...

-o enp0s3: ... y que vayan a salir (*output*) por la interfaz indicada.

-j SNAT: Indica qué acción va a activar (*jump*) esta regla. *SNAT* indica que se sustituirá la IP origen del paquete por una dirección estática (*Static NAT*).

--to X.X.X.20X: Indica la dirección por la cual debe sustituirse la IP origen. En este caso, la propia IP externa del *Servidor*.

NOTA: en caso de que la IP externa del Servidor no la conozcamos (principalmente porque tenga una IP dinámica proporcionada por un servidor DHCP), no sabríamos qué poner en el último parámetro, donde se indica la IP que se designa como origen de los paquetes.

En ese caso, podemos indicarle a la regla que consulte (que enmascare) ella misma la dirección IP, sustituyendo los dos últimos parámetros...

-j SNAT --to X.X.X.20X

... por simplemente:

-j MASQUERADE

Podemos consultar las reglas ya creadas en la tabla NAT con el mismo comando **iptables**:

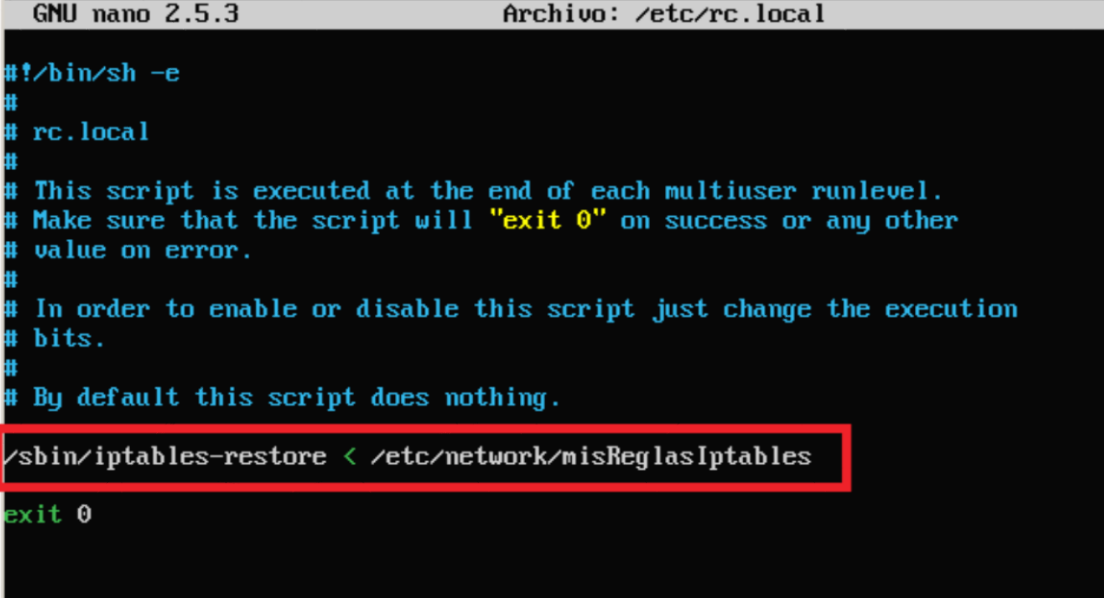
```
~$ sudo iptables -t nat -L
```

Sin embargo, estas reglas no se mantendrán tras el apagado del *Servidor*. Para asegurarnos de que nuestro dispositivo seguirá aplicando NAT en el futuro, tendremos que guardar las reglas en un archivo y asegurarnos de que ese archivo se “cargue” cada vez que el *Servidor* se ponga en marcha.

Podemos guardar las reglas actuales redirigiendo la salida del comando **iptables-save** a un archivo cualquiera que nos inventemos, por ejemplo:

```
~$ sudo iptables-save > /etc/network/misReglasIptables
```

Finalmente, podemos incluir el comando **iptables-restore** (ubicado en */sbin*) en el archivo **/etc/rc.local** para que nuestro sistema lo ejecute tras cada inicio:



```
GNU nano 2.5.3          Archivo: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

/sbin/iptables-restore < /etc/network/misReglasIptables
exit 0
```

Tendremos que reiniciar la máquina para comprobar que la regla NAT sigue apareciendo en el listado de reglas de **iptables**.

5. Comprobaciones

En este momento, tanto el *Servidor* como el *Cliente 1* deberían tener conexión a internet. Además, los servidores y clientes de cada alumno deberían poder comunicarse (*ping*) con los servidores del resto de compañeros (no así con sus clientes).

Una vez nos hemos asegurado del correcto funcionamiento, podemos instalar y configurar la máquina *Cliente 2* de exactamente la misma manera, excepto por su dirección IP. Si estamos trabajando con máquinas virtuales, puedes clonar el *Cliente 1* y modificar simplemente su dirección IP (*¡en ese caso, recuerda cambiar también la dirección física de su tarjeta de red!*).

Y eso es todo: ya tenemos una red LAN totalmente funcional sobre la cual podremos realizar pruebas de las aplicaciones y servicios que iremos viendo a lo largo del módulo.