

|  |
| --- |
| TFG del Grado en Ingeniería Informática  Simulación de redes con GNS3  Documentación Técnica |



Presentado por Javier García González

en Universidad de Burgos -- 21 de enero de 2019

Tutores: Alejandro Merino Gómez y Daniel Sarabia Ortiz

Índice general

[Apéndice A 4](#_Toc535222258)

[**A.1. Introducción** 4](#_Toc535222259)

[**A.2. Planificación temporal** 4](#_Toc535222260)

[**A.3. Estudio de viabilidad** 4](#_Toc535222261)

[Apéndice B 5](#_Toc535222262)

[**B.1. Introducción** 5](#_Toc535222263)

[**B.2. Diseño de la red** 5](#_Toc535222264)

[Apéndice C 6](#_Toc535222265)

[**C.1. Introducción** 6](#_Toc535222266)

[**C.2. Requisitos de usuarios** 6](#_Toc535222267)

[**C.3. Instalación** 6](#_Toc535222268)

[**VirtualBox (versión instalada 5.2.18)** 6](#_Toc535222269)

[**VMWare Workstation 12 player** 7](#_Toc535222270)

[**GNS3 2.1.0** 8](#_Toc535222271)

[**C.4. Manual del usuario** 9](#_Toc535222272)

[**Conectar GNS3 a redes externas (Internet)** 9](#_Toc535222273)

[**Conectar GNS3 contra un servidor remoto** 10](#_Toc535222274)

[**Configuración de redes virtuales VLAN.** 12](#_Toc535222275)

[**Servidor Web con apache sobre Ubuntu.** 14](#_Toc535222276)

[**Servidor DHCP.** 16](#_Toc535222277)

[**Servidor DHCP. Configurar retransmisor** 20](#_Toc535222278)

Índice de figuras

[Ilustración 1: Topología de la red. 5](#_Toc535222207)

[Ilustración 2: Oracle VM VirtualBox 7](#_Toc535222208)

[Ilustración 3: VMWare 7](#_Toc535222209)

[Ilustración 4: GNS3 VM 8](#_Toc535222210)

[Ilustración 5: Entorno de GNS3 9](#_Toc535222211)

[Ilustración 6: Conexión a Internet 10](#_Toc535222212)

[Ilustración 7: IP equipo Mesa 11](#_Toc535222213)

[Ilustración 8: IP equipo Portátil 11](#_Toc535222214)

[Ilustración 9: GNS3 configuración servidor remoto 11](#_Toc535222215)

[Ilustración 10: Opciones ejecución GNS3 12](#_Toc535222216)

[Ilustración 11: Redes VLAN 12](#_Toc535222217)

[Ilustración 12: Redes VLAN con administración 14](#_Toc535222218)

[Ilustración 13: Apache configuración 15](#_Toc535222219)

[Ilustración 14: Apache configuración 2 15](#_Toc535222220)

[Ilustración 15: Apache configuración 3 15](#_Toc535222221)

[Ilustración 16: Prueba Apache 16](#_Toc535222222)

[Ilustración 17: Fichero configuración DHCP 17](#_Toc535222223)

[Ilustración 18: Fichero configuración DHCP 2 18](#_Toc535222224)

[Ilustración 19: Fichero configuración DHCP3 18](#_Toc535222225)

[Ilustración 20: Topología DHCP 19](#_Toc535222226)

[Ilustración 21: Fichero configuración DHCP 4 19](#_Toc535222227)

[Ilustración 22: Retransmisor DHCP 20](#_Toc535222228)

# Apéndice A

Plan de Proyecto Software

**A.1. Introducción**

En este apartado se desglosa el estudio previo al proyecto, el tiempo dedicado y la viabilidad del proyecto.

**A.2. Planificación temporal**

Para el desarrollo de todas las partes del proyecto se ha utilizado el repositorio GitHub donde nos permite crear un proyecto y ver en cada momento que apartados tenemos pendientes, en proceso o ya realizados.

En un principio el tiempo que se ha estimado para el proyecto ha sido de aproximadamente dos horas diarias debido a que se ha tenido que compaginar con el trabajo. De esta forma se ha destinado al proyecto 2 horas diarias los días laborables, ya que la jornada finalizaba a las 19horas. Los fines de semana, dependiendo de las fechas, se ha destinado al proyecto un tiempo de 4-5 horas diarias.

Teniendo en cuenta el tiempo disponible se ha adaptado el proyecto modificando sobre la marcha alguno de los objetivos, centrando más el desarrollo en algunos puntos considerados más interesantes y dejando de lado aquellos que bien por no tener tiempo suficiente para realizarlos íntegramente o haber sido ya estudiados en las asignaturas cursadas a lo largo del grado.

**A.3. Estudio de viabilidad**

Durante el desarrollo del proyecto se ha ido estudiando la viabilidad de muchos de los apartados, dejando algunos de ellos de lado por no ser posible realizarlos con las herramientas de las que disponíamos, este es por ejemplo el caso de la configuración remota de la red. Hemos centrado este caso de estudio en la configuración remota de router y equipos con Ubuntu ya que con el resto de los elementos no nos ha sido posible.

Una vez se ha demostrado la viabilidad del resto de objetivos, bien antes o bien durante su realización, se ha comprobado el funcionamiento de los elementos y se ha llevado a cabo su estudio.

# Apéndice B

Especificación de diseño

## **B.1. Introducción**

En este apartado tratamos el tema del diseño de la red a construir, previamente a comenzar a unir todas partes, se han generado todas ellas por separado, de esta forma hemos podido hacer pruebas más simples sobre ellas y también hemos podido hacer un diseño y estudio de la topología de la red final.

## **B.2. Diseño de la red**

Una vez decididos todos los componentes que van a formar parte de la red se realiza topología que seguirá de forma que todo quede conectado y pueda funcionar de forma óptima:

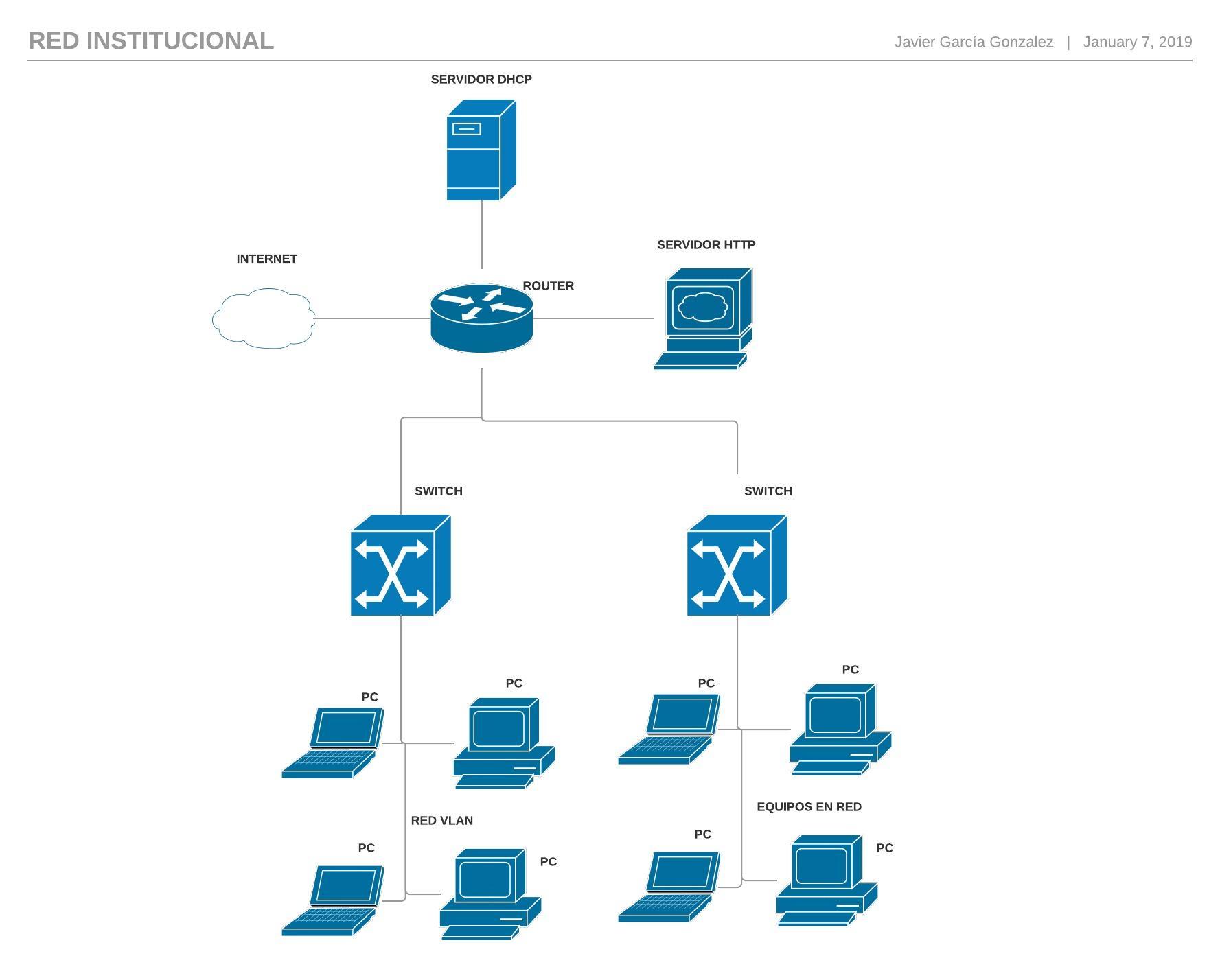


Ilustración 1: Topología de la red.

Apéndice C

Documentación de usuario

## **C.1. Introducción**

En este apartado veremos que necesitamos para poder utilizar el proyecto y poder realizar mejoras, pruebas o cualquier modificación sobre el código entregado.

## **C.2. Requisitos de usuarios**

Los requisitos para poder utilizar el software visto en este proyecto son los siguientes:

* Disponer de al menos dos equipos con sistema operativo Windows.
* Conexión a Internet.
* Un router que nos permita tener los dos equipos en red.
* Espacio suficiente para la instalación de todo el software. Esto depende del número y tipo de máquinas virtuales instaladas, en el caso del proyecto tal cuál, son necesarios al menos 10 GB para garantizar el correcto funcionamiento.

## **C.3. Instalación**

A continuación, se muestra el proceso a seguir para poder tener el proyecto completo con todo el software requerido.

### **VirtualBox (versión instalada 5.2.18)**

En primer lugar, para comenzar con el proyecto se ha instalado el software de virtualización VirtualBox en su versión más reciente a la fecha (5.2.18). Se trata de un programa gratuito desarrollado por Oracle. Gracias a ello podemos ejecutar el sistema operativo Ubuntu Server (en su versión 14.04.4 LTS Server). Podemos cargar estas máquinas más adelante en nuestro proyecto para poder simular elementos de la red. En la instalación inicial se han cargado dos máquinas virtuales de Ubuntu para futuros usos, que según ha ido creciendo el proyecto se han unido más máquinas, todas ellas con Ubuntu:

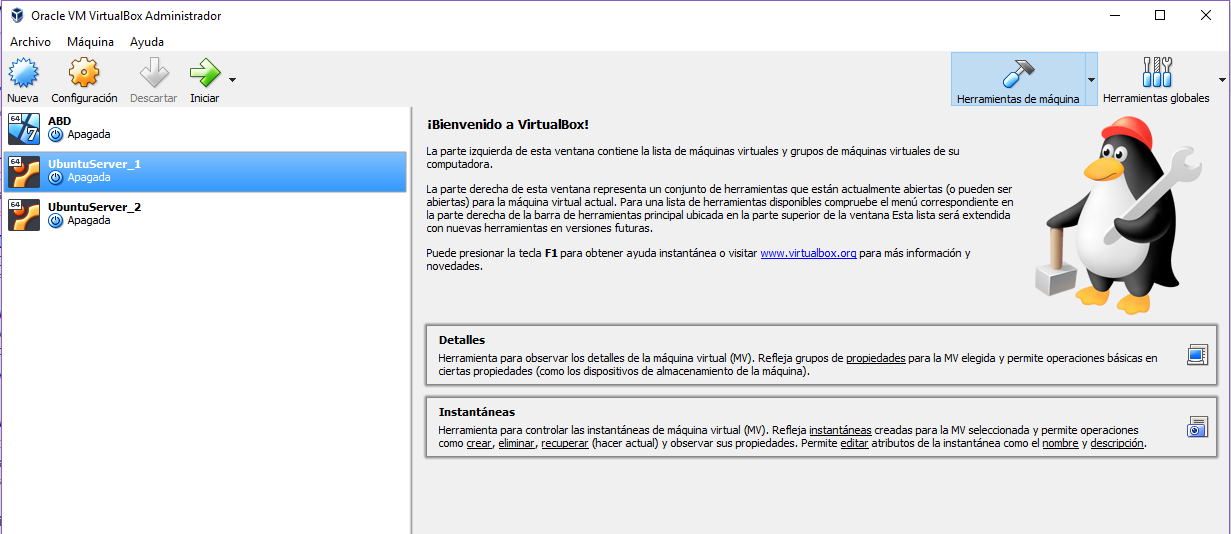


Ilustración 2: Oracle VM VirtualBox

### **VMWare Workstation 12 player**

Otro de los software que he necesitado para realizar el proyecto es VMWare en su versión Workstation 12 player (12.5.8). VMWare pertenece a la empresa EMC Corporation (Dell Inc), nos proporciona este software de virtualización totalmente gratis. Una vez instalado vamos a cargar dentro una virtualización del programa que veremos más adelante y en el cual se basa el proyecto: GNS3. Para poder conectar ambos programas deben estar en la misma versión, que es la versión 2.1.0.

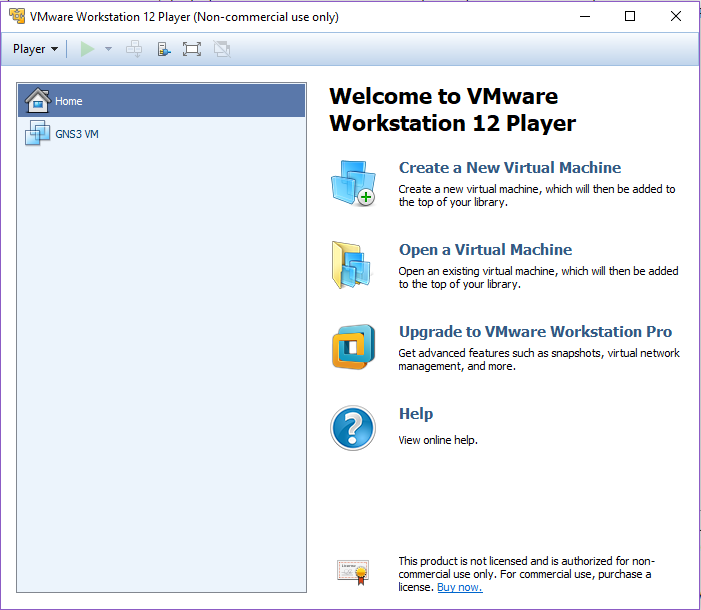


Ilustración 3: VMWare

### **GNS3 2.1.0**

Se trata del programa principal sobre el que se desarrolla este proyecto. Es un software gratuito que permite la simulación gráfica de topologías de red muy complejas y su ejecución. Este programa funciona gracias a Dynamips, se trata de un emulador de sistemas operativos que nos permite ejecutar binarios de Cisco entre otros muchos.

Para su funcionamiento se sincroniza con VMWare y la máquina de GNS3 antes importada sobre este. De esta forma evitamos realizar toda la carga de recursos del programa sobre nuestro propio equipo y lo hacemos sobre el sistema de GNS3 de la máquina virtual. Si se ha realizado correctamente, nada más iniciar GNS3 se nos cargará la maquina de VMWare:

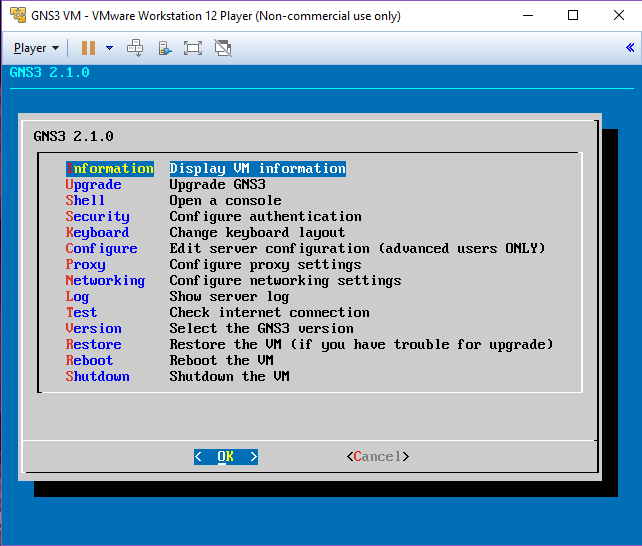


Ilustración 4: GNS3 VM

El entorno que nos encontramos nada más abrir el programa será el siguiente:

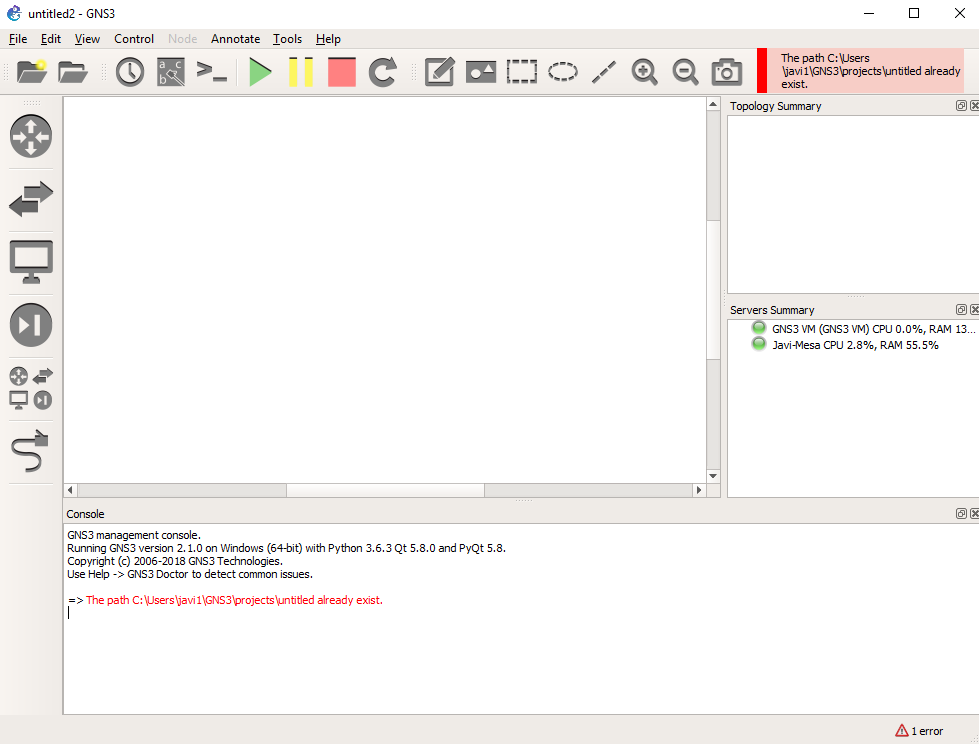


Ilustración 5: Entorno de GNS3

Como podemos ver en la parte derecha ya podemos escoger el sistema sobre el que trabajar, en primer lugar, vemos el sistema sobre GNS3 de VMWare y en segundo lugar tenemos el equipo local. Ya estamos preparados para comenzar a utilizar el programa.

## **C.4. Manual del usuario**

Vemos la realización de todos los objetivos por separado de forma que nos permite llevar un manual de instalación de todos los componentes. A continuación, vemos el proceso a seguir:

### **Conectar GNS3 a redes externas (Internet)**

Para la red queremos tener un punto de acceso a través del cual tener acceso con el exterior y poder conectar a internet. Para ello utilizamos un Router que con una configuración sencilla gracias a la potencia de GNS3 vamos a poder tener conexión a Internet.

Vamos a necesitar únicamente dos componentes, el mencionado Router y un elemento llamado cloud de GNS3.

La configuración necesaria es la que sigue:

En primer lugar, en el equipo vamos a las conexiones de red y entramos a las propiedades de red del adaptador que esté funcionando en nuestro caso, permitimos la conexión de otras redes a través de este equipo, esto nos permitirá escogerlo para nuestra red en GNS3

La nube deberemos configurarla para que su conexión Ethernet adquiera las propiedades de la tarjeta de red a través de la cuál nuestro equipo obtiene la conexión a Internet compartida del paso anterior, de esta forma tendrá acceso al adaptador que ofrece Internet al equipo. En este caso es el adaptador Ethernet.

En el router (una vez lo hemos conectado a la nube y encendido) tan solo debemos activar el enlace utilizado, darle una dirección IP dentro de la subred de nuestro equipo y activar los comandos necesarios:

Damos al puerto asignado (f0/0) una dirección IP cualquiera de la red, para ello podemos utilizar DHCP, asi adquiere automáticamente una IP:

ip address dhcp

ip domain-lookup

Una vez seguidos estos pasos podremos hacer ping a cualquier web:

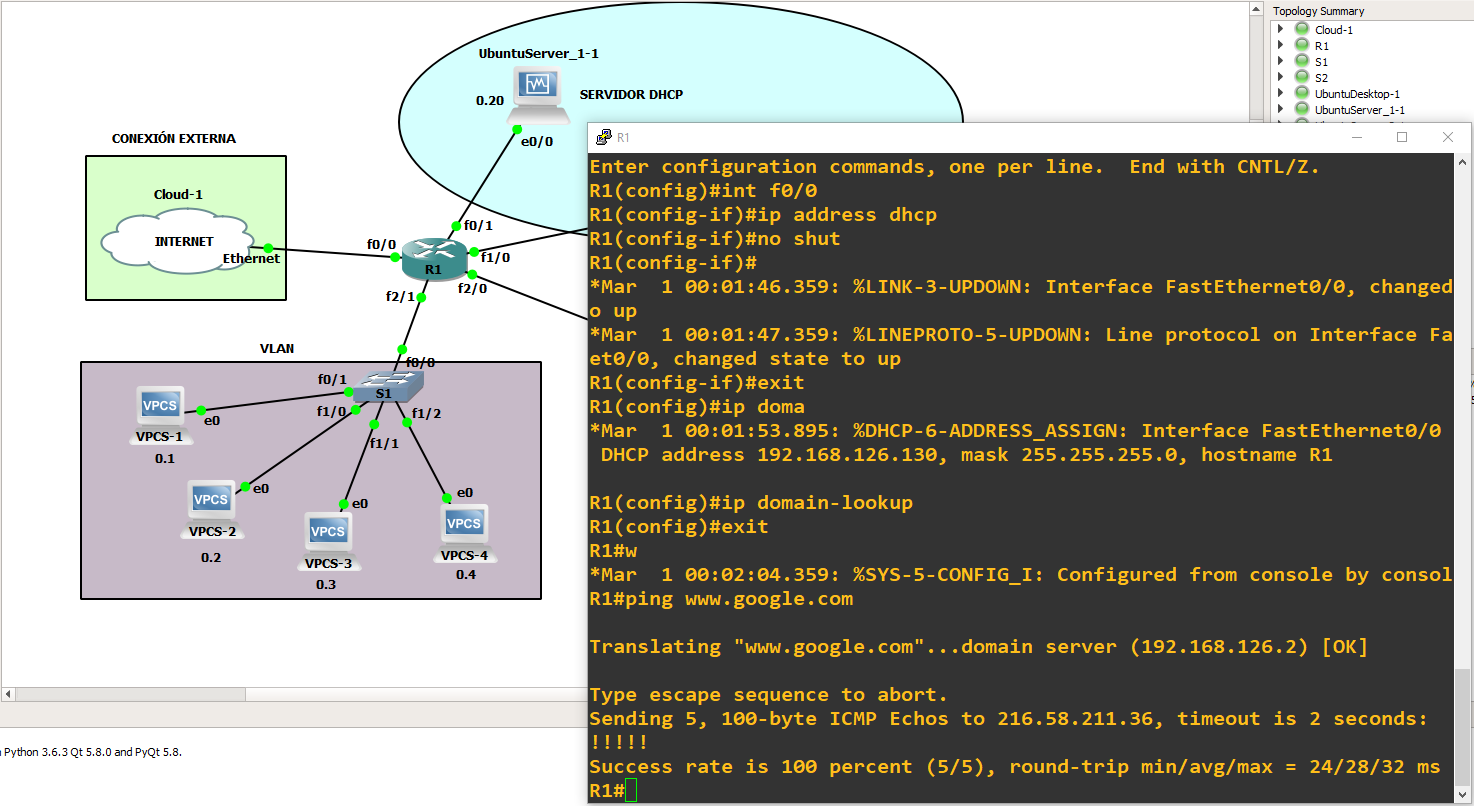


Ilustración 6: Conexión a Internet

### **Conectar GNS3 contra un servidor remoto**

He realizado el proceso de lanzar GNS3 en uno de mis equipos (portátil) contra otro equipo (mesa). Este proceso puede ser muy útil para casos en los que nuestra red empiece a crecer y un equipo carezca de recursos, de esta forma conseguimos que la carga de trabajando sea sobre el servidor y así el equipo en el que estamos trabajando sufra menos de esa carga.

El proceso es sencillo, una vez tenemos la instalación en los dos equipos como hemos visto en los pasos anteriores de instalación, nos fijamos en la IP que tenga cada equipo.

En mi caso en el equipo de mesa (que en este ejemplo hace de servidor):

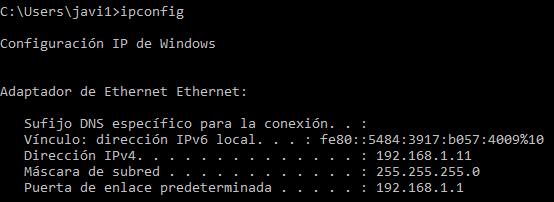


Ilustración 7: IP equipo Mesa

Y en el portátil:

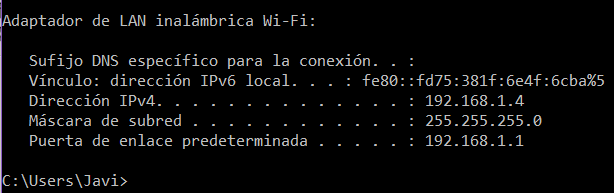


Ilustración 8: IP equipo Portátil

Una vez tenemos claro esto, en la instalación GNS3 del portátil, vamos a cambiar la forma de conectar el programa, desde la opción Setup Wizard:

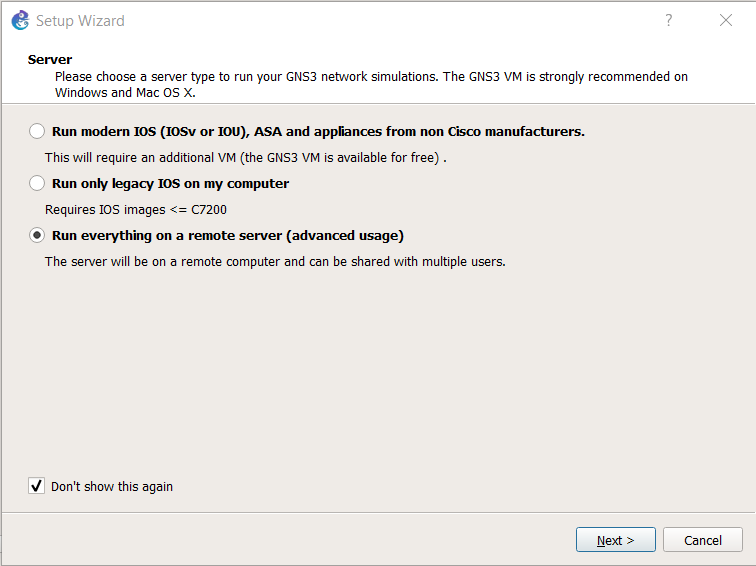


Ilustración 9: GNS3 configuración servidor remoto

Escogemos la opción de ejecutar sobre servidor remoto, como en la imagen.

Ahora debemos poner la dirección del servidor, para ello hemos buscado la IP de cada equipo, de esta forma ahora podemos introducirla, desmarcamos la opción de autentificación ya que nadie más va a tener acceso y el resto lo dejamos con las opciones por defecto.

En este último paso ya solo debemos confirmar que esta todo correcto, el programa conectará al servidor y podremos pulsar en finalizar, el resultado es el siguiente:

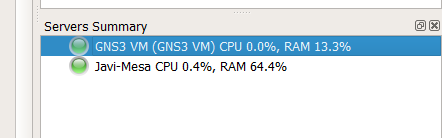


Ilustración 10: Opciones ejecución GNS3

Ya tenemos configurados nuestros equipos para que se ejecute GNS3 en remoto sobre el servidor.

### **Configuración de redes virtuales VLAN.**

Para configurar redes virtuales vamos a utilizar un router con funciones de Switch como el que hemos creado en la instalación del programa. Para ello vamos a dividir la red en dos subredes virtuales:

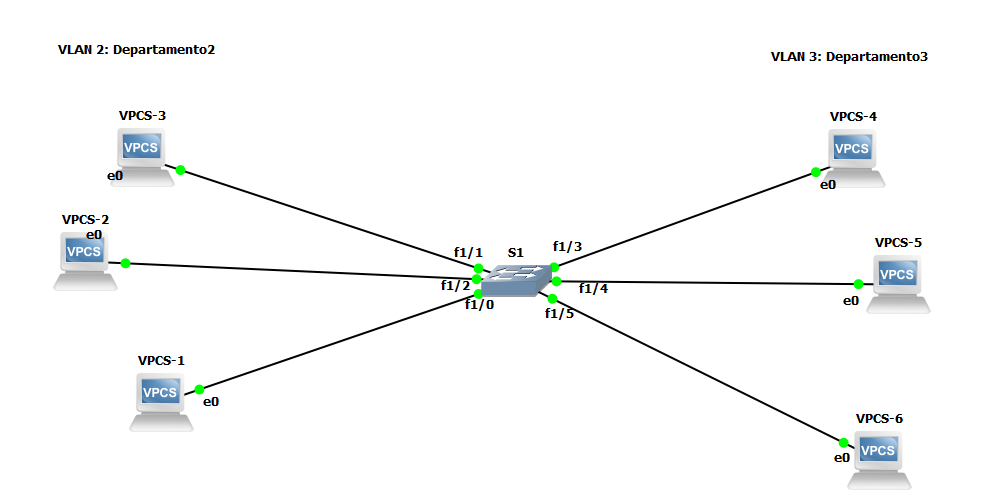


Ilustración 11: Redes VLAN

A partir de esta red, empezamos a configurar los equipos, le damos una IP a cada uno coincidiendo con su nombre, el VPCS-1 tendrá la IP 192.168.0.1, el 2 lo mismo, pero con la IP 192.168.0.2 y así sucesivamente. Esto lo hacemos mediante el comando:

Ip 192.168.0.x/24

Tras esto pasamos a configurar el Switch, activando todos los puertos a los que se conectan el resto de los equipos, utilizamos para ello:

Configure terminal  
Interface f1/x  
No shutdown

Una vez configurados los equipos y activados los enlaces vamos a crear las VLAN:

configure terminal

VLAN 2  
name Departamento2

Ahora tendremos que asignar a cada puerto la VLAN a la que queramos que pertenezca:

interface f1/x

switchport access vlan 2

interface f1/y

switchport access vlan 3

Podemos ver la configuración de cada VLAN y que puertos tiene asignados utilizando:

Show vlan-sw

En el proyecto completo vamos a tener los elementos:

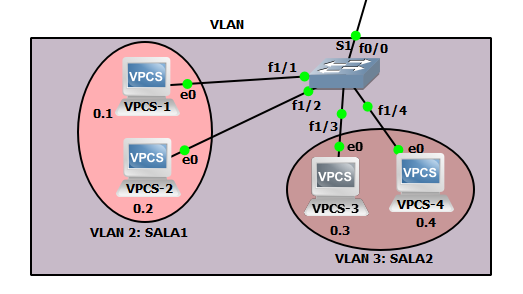


Ilustración 12: VLAN

Y probamos el acceso entre misma VLAN y vemos que falla con la red externa.

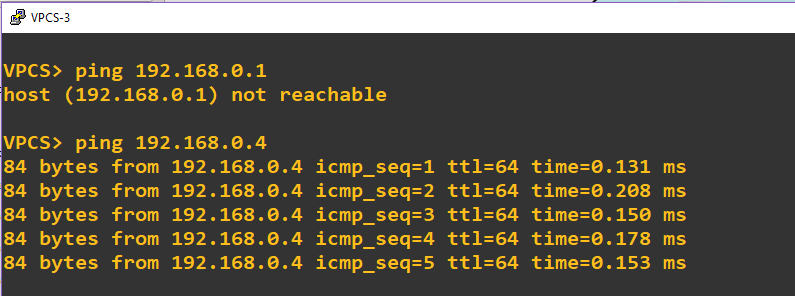


Ilustración 13: Prueba VLAN

Ahora vamos a crear una VLAN de administración para poder gestionar el Switch. En primer lugar, vamos a configurar las conexiones telnet para permitir múltiples conexiones:

line vty 0 4

Creamos una contraseña de acceso:

pass 123456

login

Definimos la VLAN para administración y le damos una IP:

vlan 99

name admin

interface vlan 99

ip address 192.168.20.20 255.255.255.0

no shutdown

Incluimos la interface que vayamos a utilizar del Switch para administración en la VLAN:

interface FastEthernet1/15

switchport access vlan 99

Una vez tenemos la VLAN de administración lista, vamos a unir a ese puerto un Ubuntu server desde el cual poder conectarnos:

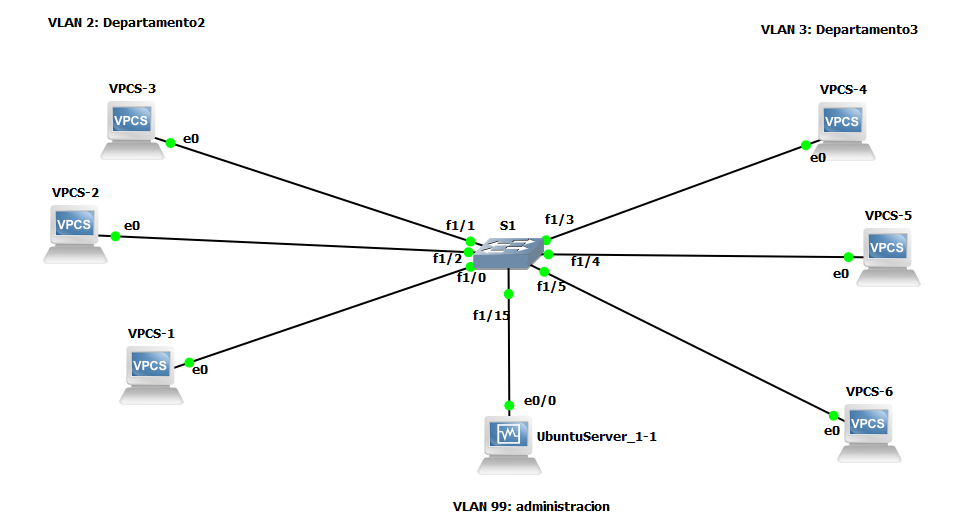


Ilustración 14: Redes VLAN con administración

### **Servidor Web con apache sobre Ubuntu.**

1. Preparación de Ubuntu.

En primer lugar, vamos a configurar un cortafuegos que nos permite bloquear aquellos puertos que no vamos a necesitar, vamos a utilizar UFW. En principio únicamente queremos activarlo para utilizarlo más adelante:

Sudo ufw enable

1. Actualización de Ubuntu.

A continuación, realizo la actualización de Ubuntu con:

Sudo apt update

1. Instalación de Apache.

Tras la actualización, pasamos a la instalación de apache2, para ello vamos a ejecutar el comando:

sudo apt install apache2

esto nos pedirá varias confirmaciones y nos permitirá utilizar apache sobre nuestro Ubuntu.

1. Configurar el cortafuegos.

Vamos a ver las aplicaciones que tenemos:

Sudo ufw app list

Permitimos la conexión de apache a través del cortafuegos:

Sudo ufw allow Apache



Ilustración 15: Apache configuración

Y comprobamos el estado del cortafuegos:

Sudo ufw status

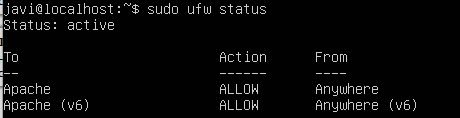


Ilustración 16: Apache configuración 2

Una vez tenemos el cortafuegos configurado podemos verificar que el servicio de Apache esta activo:

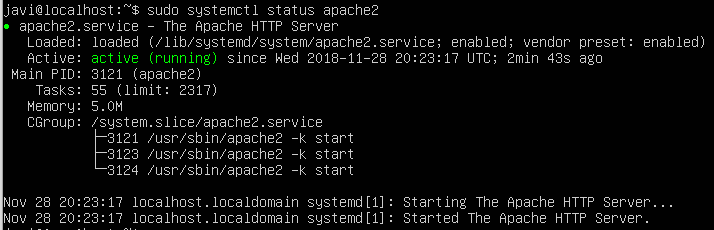


Ilustración 17: Apache configuración 3

Vamos a crear un sitio virtual para probar la conexión con el resto de los equipos. Para ello creamos un directorio dentro del bloque que trae por defecto apache:

sudo mkdir -p /var/www/ejemplo.com/html

Asignamos propietario al directorio:

sudo chown -R $USER:$USER /var/www/ejemplo.com/html

Cambiamos los permisos sobre el directorio para asegurarnos de que este correcto:

sudo chmod -R 755 /var/www/ejemplo.com

Vamos a crear una página de ejemplo tfg.html:

nano /var/www/ejemplo.com/html/tfg.html

Para poder visualizar este contenido lo tenemos que guardar en un espacio virtual con sus directivas. Vamos a crear este espacio:

sudo nano/etc/apache2/sites-available/ejemplo.com.conf

Y lo creamos con lo siguiente:

<VirtualHost \*:80>

ServerAdmin admin@ejemplo.com

ServerName ejemplo.com

ServerAlias www.ejemplo.com

DocumentRoot /var/www/ejemplo.com/html

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Ahora necesitamos habilitar el archivo que acabamos de guardar:

sudo apache2ctl configtest

Reiniciamos el servicio apache para que los cambios surtan efecto:  
sudo systemctl restart apache2

Ahora configuramos una red simple para hacer las pruebas y tratamos de llegar desde el cliente al servidor http. Accedemos utilizando el navegador links:

links http://192.168.0.2

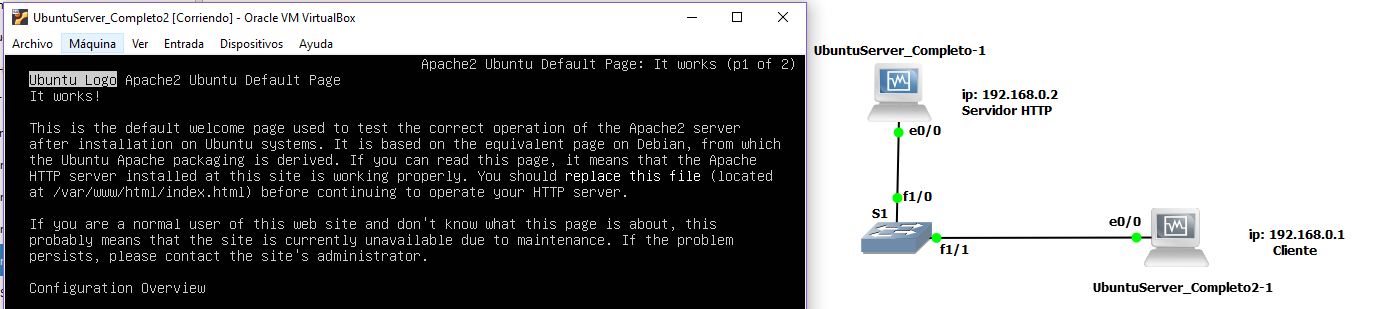


Ilustración 18: Prueba Apache

### **Servidor DHCP.**

Vamos a configurar una máquina virtual con Ubuntu de forma que pueda utilizarse como un servidor DHCP.

En primer lugar, vamos a permitir la conexión a Internet de la máquina virtual, para ello vamos a la configuración de la máquina en VirtualBox y cambiamos el adaptador a NAT y habilitamos la opción cable conectado.

Una vez se tiene acceso a Internet, instalamos el servidor DHCP con el comando:

sudo apt-get install isc-dhcp-server

Tras la instalación quitamos los cambios del adaptador de red, sino nos dará error al tratar de utilizar la máquina en GNS3.

Ahora vamos a darle al servidor una dirección IP:

sudo nano /etc/network/interfaces

Y dejamos el fichero de la siguiente forma:

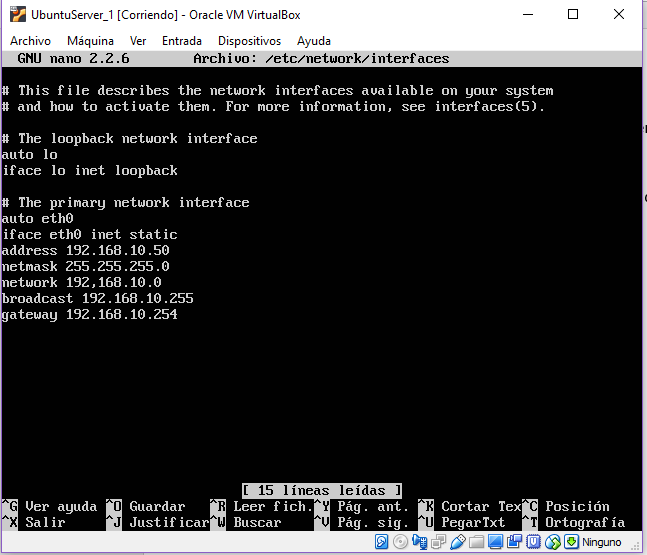


Ilustración 19: Fichero configuración DHCP

Como el servidor no puede asignarse una IP a sí mismo, por eso le damos la dirección estática a mano fuera del rango que dará a los clientes.

Ahora modificamos las interfaces por las cuales va a poder recibir peticiones. Modificamos el archivo:

sudo nano /etc/default/isc-dhcp-server

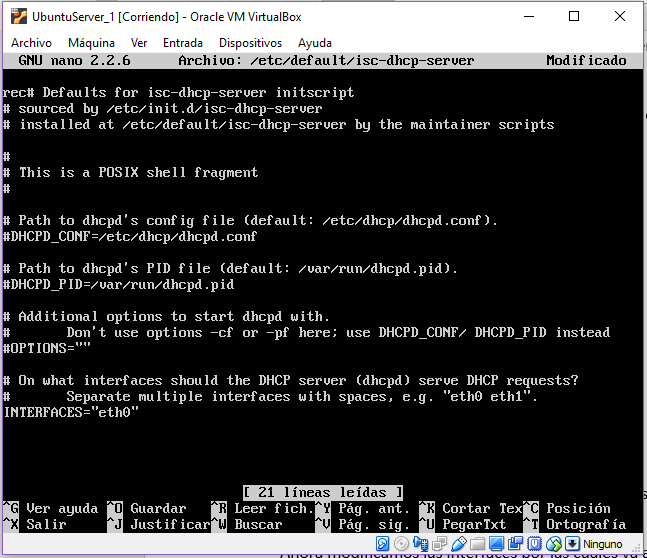


Ilustración 20: Fichero configuración DHCP 2

Ahora vamos a configurar el servidor DHCP en el archivo:

sudo nano /etc/dhcp/dhcpd.conf

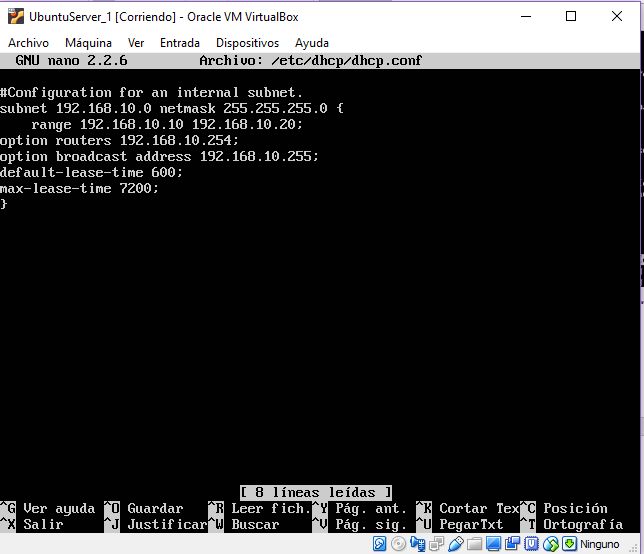


Ilustración 21: Fichero configuración DHCP3

Con esto solo nos queda probar, creamos la red que se muestra a continuación:

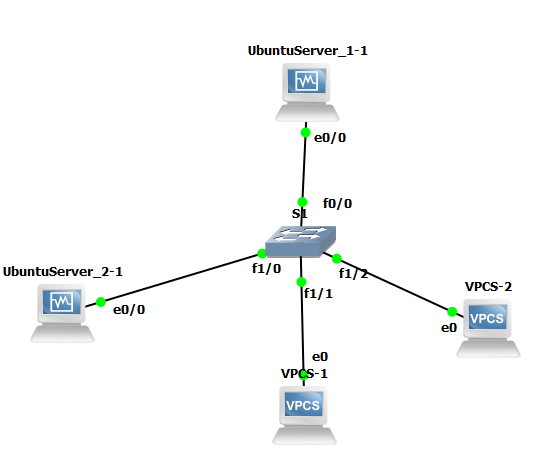


Ilustración 22: Topología DHCP

En los equipos con Ubuntu, para obtener la dirección mediante el servidor DHCP tenemos que configurar la interfaz, al igual que en el servidor, modificando el fichero /etc/network/interfaces:



Ilustración 23: Fichero configuración DHCP 4

Para liberar la dirección se usará el comando:

sudo dhclient -r eth0

Y para cambiar la actual por otra:

sudo dhclient eth0

En los VPCs configurados el proceso es más sencillo aún, tan solo debemos escribir por consola:



Para liberar la conexión:



Para renovar la dirección:



### 

### **Servidor DHCP. Configurar retransmisor**

Si nos encontramos en el caso de que el servidor DHCP no se encuentra en la misma subred que los equipos conectados, es necesario el uso de routers retransmisores, si no hacemos esto, los routers no reenvían los mensajes de difusión.

Esta es la razón por la que vamos a configurar un router como retransmisor DHCP o DHCP relay. Para que un router se comporte como DHCP relay vamos a especificar la interfaz de la subred en la que se encuentran los equipos y configurarlo para que la interfaz encuentre la IP del servidor DHCP.

Esto es muy sencillo, tan solo vamos a añadir el router entre la red y el servidor de la siguiente forma:

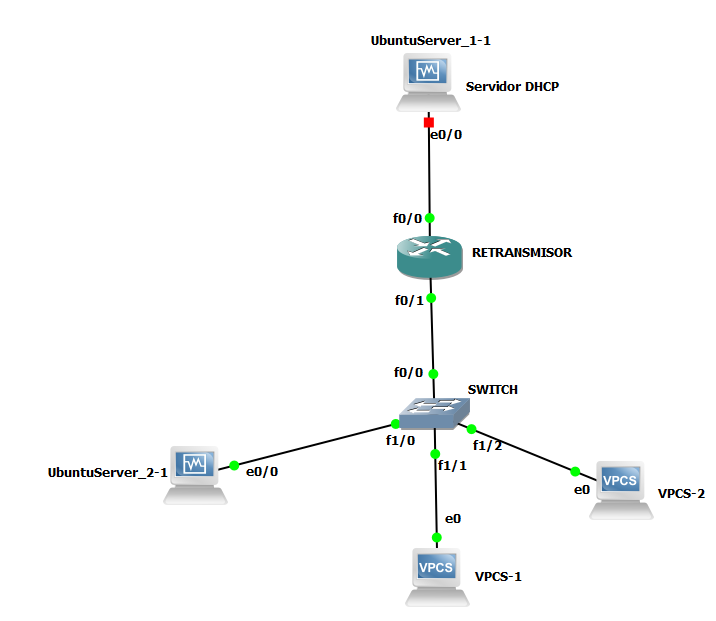


Ilustración 24: Retransmisor DHCP

En el router retransmisor vamos a hacer la configuración con los comandos:

enable

configure terminal

interface FastEthernet0/0

ip helper-address 192.168.0.20

### **Enrutamiento dinámico**

Una vez tenemos la red final configurada con todos los elementos, debemos unirlos a través de un router que establezca los caminos. Para este caso vamos a emplear el enrutamiento dinámico RIP, el cual es capaz de establecer las rutas óptimas indicando todas las subredes conectadas.

Para ello necesitamos eliminar cualquier enrutamiento estático que se haya configurado previamente, podemos ver todas las rutas y después eliminarlas con los parámetros del router:

show ip route

configure terminal

no ip route 145.66.0.0 255.255.0.0

Vemos ahora las redes que debemos conectar a nuestro router:

192.168.0.0

192.168.2.0

192.168.10.0

192.168.20.0

Una vez las tenemos empleamos los comandos para activar el enrutamiento RIP (Donde x va a ser sustituido por cada subred que tenemos en nuestra red:

router rip

network 192.168.x.0

### **Configuración remota**

Para realizar la configuración remota de los dispositivos de la red, vamos a utilizar Python, este lenguaje de programación nos permite utilizar Telnet,