

P4-CONMUTADORES

Alejandro Rodríguez Rojas
Luis Vázquez Alejo
Miguel Córdoba Aranda
Manuel Durán Muñoz

Índice

1 Introducción.....	3
VLAN.....	3
2 Conectándose al Switch.....	4
3 Cambio de IP.....	5
4 Cambio de IP Maquinas.....	6
5 Creación de VLANs básicas.....	6
6 Creación de VLANs Asimétricas.....	9
7 Enlace entre VLANs.....	11
8 Equipos en múltiples VLANs.....	14
9 Conclusión.....	16

1 Introducción

Vamos a realizar el siguiente ejercicio:

VLAN

Configuración básica:

Crea tres VLANs 802.1q con la siguiente configuración:

- Empleados: id 10: puertos 1-5
- Estudiantes: id 20: puertos 6-10
- Administración: id 30: puertos 11-15

1. Conecta dos equipos al conmutador en dos puertos de la VLAN de empleados. Asígnales direcciones válidas en el rango 192.168.10.0/24. Verifica su conectividad.
2. Cambia el segundo equipo a las VLANs de estudiantes y administración. Verifica que no es posible conectar con el equipo 1.

VLAN asimétrica

1. Conecta un tercer equipo (servidor) al puerto 20 y configura su dirección IP.
2. Configura el conmutador para que los equipos 1 y 2, conectados a las VLANs de estudiantes y administración respectivamente, puedan comunicarse con él.

Enlace entre VLANs

1. Conecta dos conmutadores configurados como en el caso anterior, empleando el puerto 49.
2. Configura ambos conmutadores para que haya conectividad entre equipos conectados a la misma VLAN.

Equipos en múltiples VLANs

Modifica la configuración del tercer equipo para que tenga acceso directo a cada VLAN a través de un único interfaz de red. Asígnale direcciones distintas en cada VLAN:

- Empleados: 192.168.10.0/24
- Estudiantes: 192.168.20.0/24
- Administración: 192.168.30.0/24

1. Configura adecuadamente el puerto 30 y conecta este equipo.
2. Comprueba su conectividad con equipos conectados a las tres VLANs , configurados con la IP adecuada.

2 Conectándose al Switch

Debemos enchufar el Switch en la red eléctrica y presionar el botón de Factory Default(Altamente recomendado usar un clip para volver a los valores de fábrica).

En este caso estaremos utilizando el Switch marca Netgear de modelo FS750T2.

Una vez realizado estas tareas, la IP de configuración del Switch será la 192.168.0.239.

Para conectarnos a la interfaz, tendremos que añadir el rango de IP en nuestra máquina.

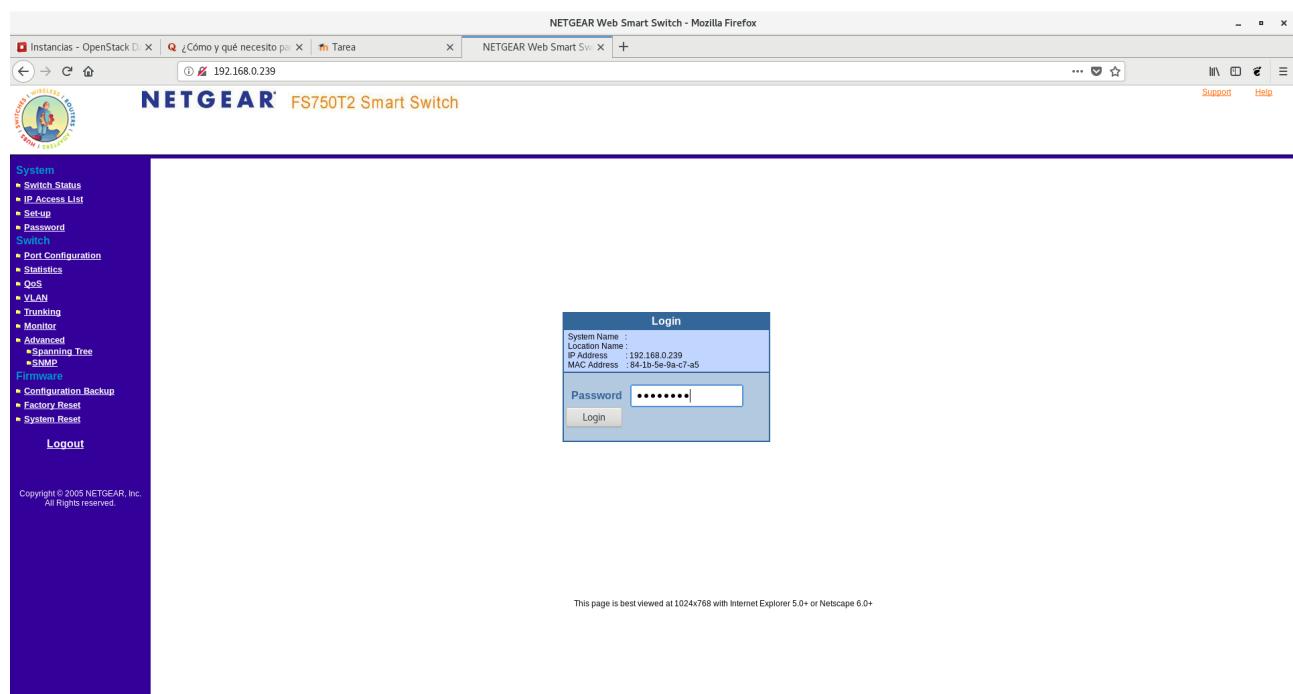
Para ello utilizaremos el comando:

```
ip addr add {IP} dev {Interfaz}
```

```
alexrr@pc-alex:~$ sudo ip addr add 192.168.0.200/24 dev enp7s0
[sudo] password for alexrr:
alexrr@pc-alex:~$ ping 192.168.0.239
PING 192.168.0.239 (192.168.0.239) 56(84) bytes of data.
64 bytes from 192.168.0.239: icmp_seq=1 ttl=255 time=0.73 ms
64 bytes from 192.168.0.239: icmp_seq=2 ttl=255 time=0.34 ms
64 bytes from 192.168.0.239: icmp_seq=3 ttl=255 time=68.8 ms
64 bytes from 192.168.0.239: icmp_seq=4 ttl=255 time=6.34 ms

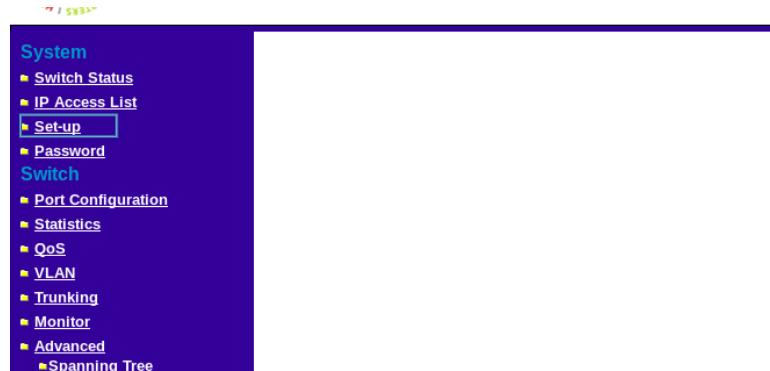
```

*Muy recomendado apagar network-manager antes de realizar el ejercicio
/etc/init.d/network-manager stop



3 Cambio de IP

Para cambiar de IP debemos usar la opción de Set-Up



Y insertar la IP estática con máscara /24

The screenshot shows the 'Static IP Address' configuration page. On the left, there is a sidebar with the following options:

- VLAN
- Trunking
- Monitor
- Advanced
- Spanning Tree
- SNMP
- Firmware
- Configuration Backup
- Factory Reset
- System Reset

Below the sidebar is a 'Logout' link and a copyright notice: 'Copyright © 2005 NETGEAR, Inc. All Rights reserved.'

The main configuration area has two radio button options:

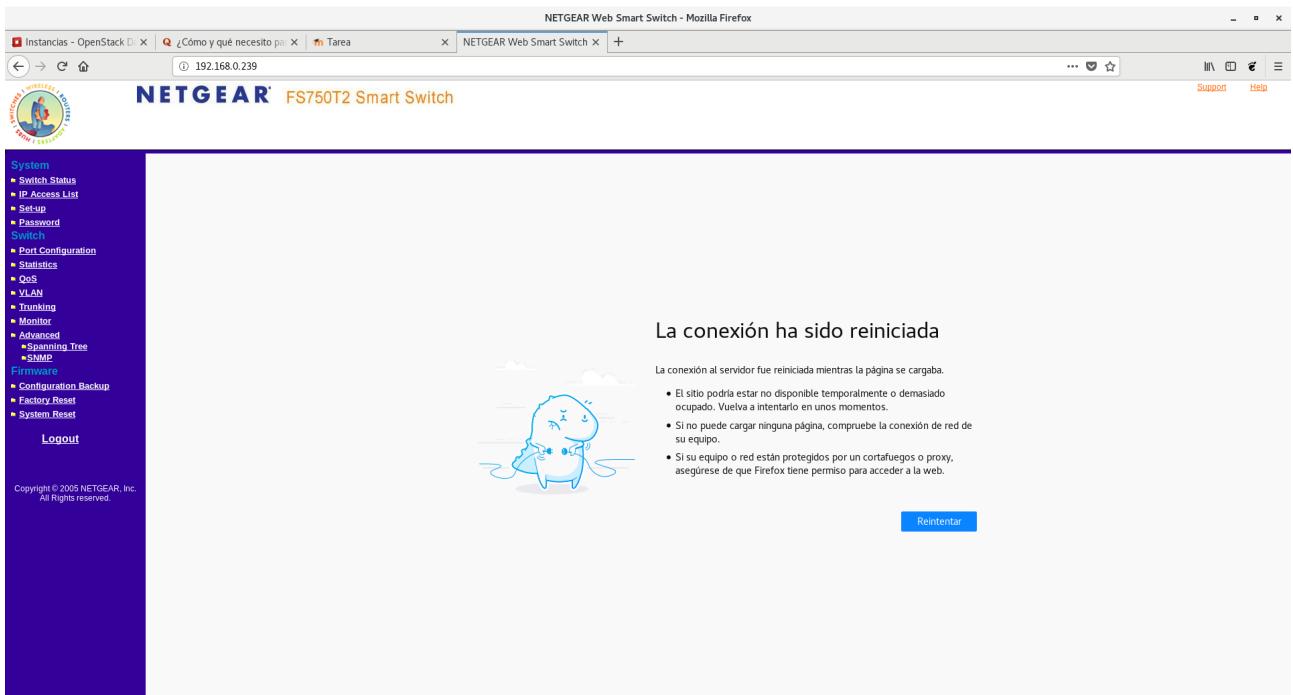
- Get Dynamic IP from DHCP Server
- Static IP Address

Below these options are three sets of input fields for IP address, Subnet mask, and Gateway, each divided into four octets by dots. The 'Static IP Address' section is selected:

IP address	10	90	10
	101		
Subnet mask	255	255	255
	0		
Gateway	10	90	10
	0		

At the bottom right are 'Apply' and 'Help' buttons.

Acto seguido el Switch se pondrá a configurar la IP.

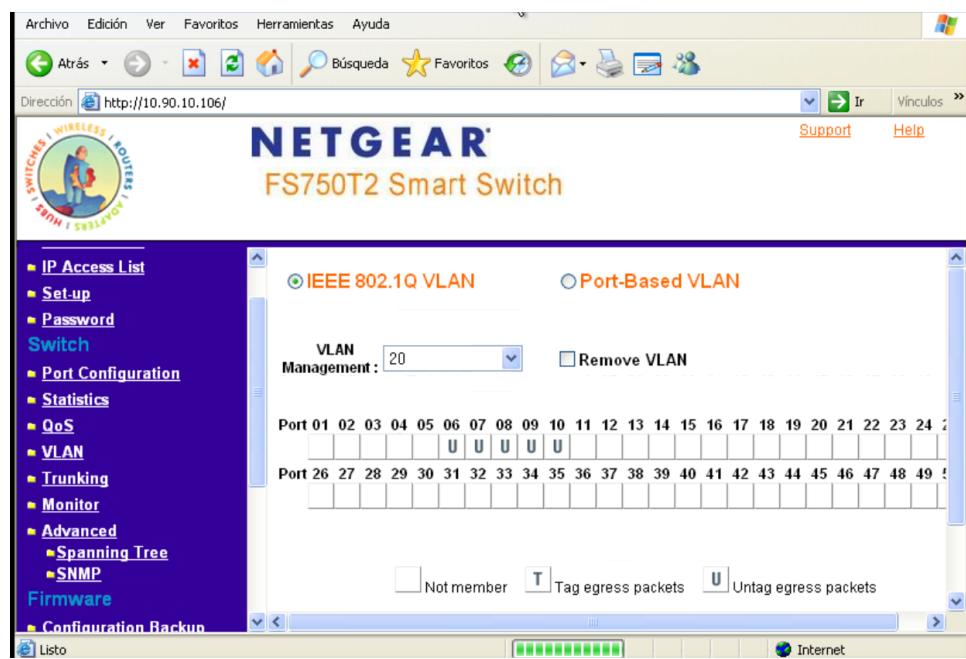
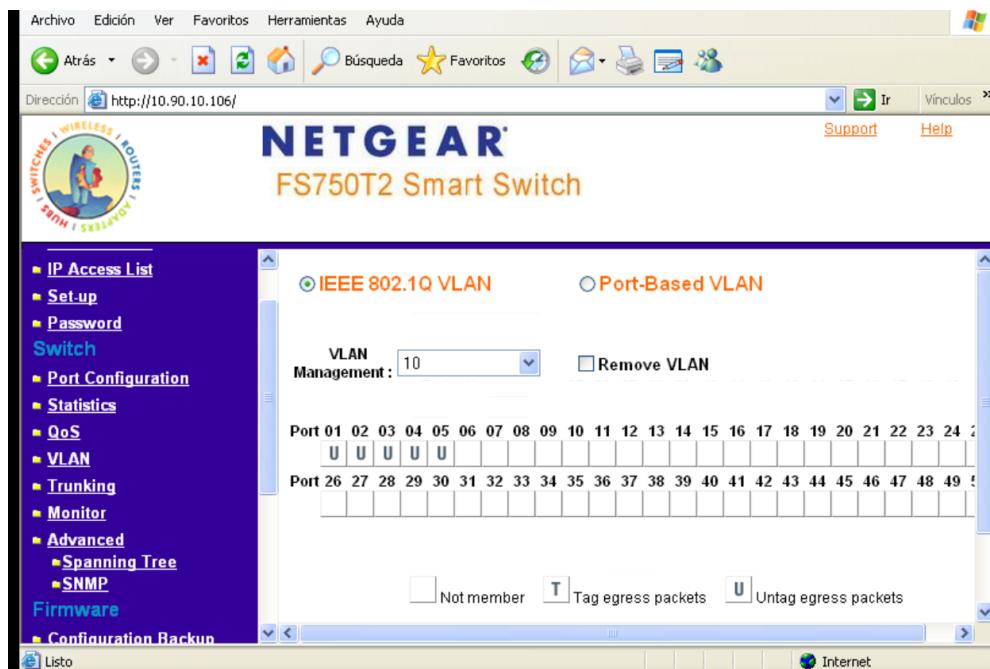


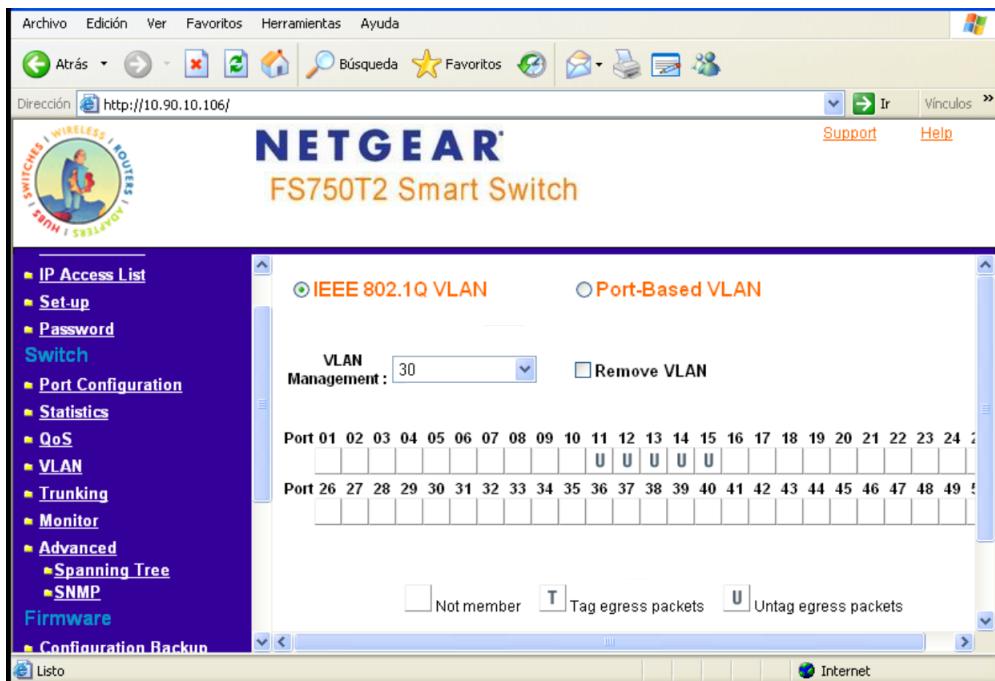
4 Cambio de IP Maquinas

Para hacernos ping entre nosotros debemos ponernos todos en el mismo rango de IP, en este caso 10.90.10.x/24.

5 Creación de VLANs básicas

Para crear las Vlan nos vamos al apartado de VLAN y seleccionamos el protocolo IEEE 802.1Q





- Conecta dos equipos al comutador en dos puertos de la VLAN de empleados. Asigna direcciones válidas en el rango 192.168.10.0/24. Verifica su conectividad.

```
alexrr@pc-alex:~$ ping 10.90.10.6
PING 10.90.10.6 (10.90.10.6) 56(84) bytes of data.
64 bytes from 10.90.10.6: icmp_seq=1 ttl=64 time=0.681 ms
64 bytes from 10.90.10.6: icmp_seq=2 ttl=64 time=0.357 ms
64 bytes from 10.90.10.6: icmp_seq=3 ttl=64 time=0.422 ms
64 bytes from 10.90.10.6: icmp_seq=4 ttl=64 time=0.404 ms
```

Verificamos que la conectividad es efectiva entre la VLAN creada

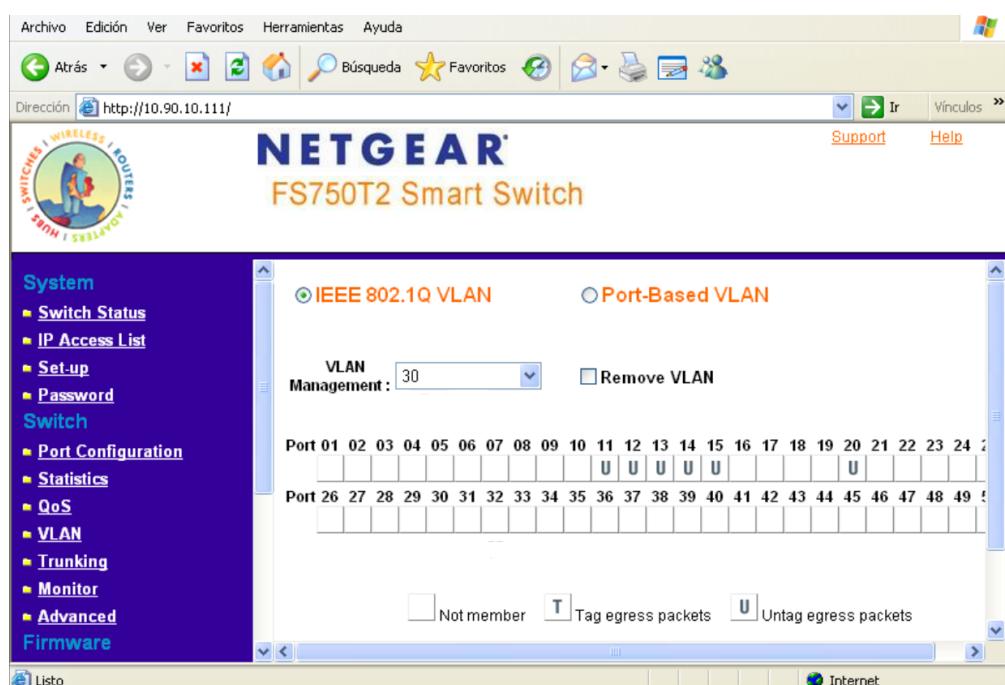
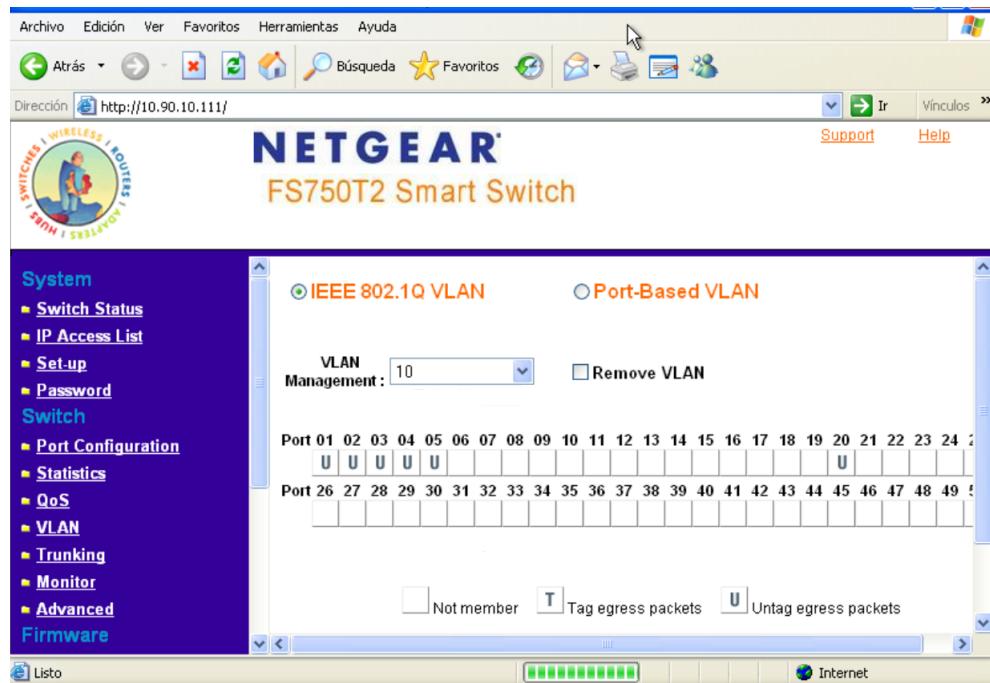
Cambia el segundo equipo a las VLANs de estudiantes y administración. Verifica que no es posible conectar con el equipo 1.

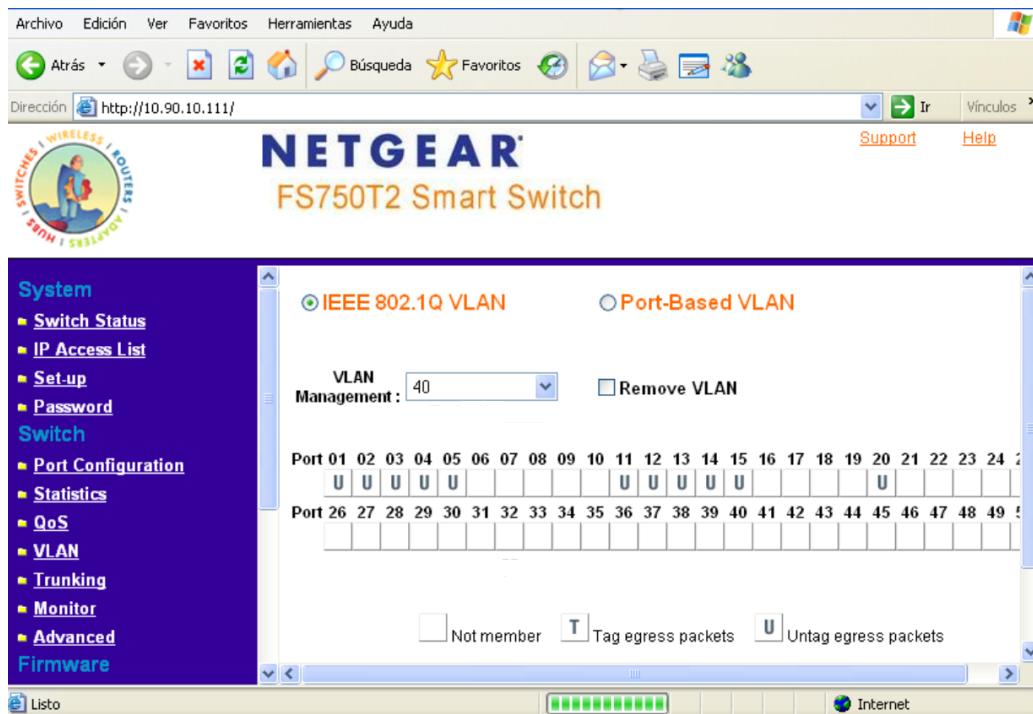
```
alexrr@pc-alex:~$ ping 10.90.10.6
PING 10.90.10.6 (10.90.10.6) 56(84) bytes of data.
```

Confirmamos que la VLAN está bien formada y no podemos realizar ping entre ambos equipos.

6 Creación de VLANs Asimétricas

Añadimos las respectivas VLANs y le añadimos el puerto 20 a dichas VLANs y también añadimos una VLAN con el puerto 20 con los puertos correspondientes.





Y comprobamos que efectivamente ambas VLAN pueden comunicarse con el puerto 20, pero no entre sí.

```

luis@kutulu:~$ ping -c 5 10.90.10.104
PING 10.90.10.104 (10.90.10.104) 56(84) bytes of data.
64 bytes from 10.90.10.104: icmp_seq=1 ttl=64 time=0.980 ms
64 bytes from 10.90.10.104: icmp_seq=2 ttl=64 time=1.21 ms
64 bytes from 10.90.10.104: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 10.90.10.104: icmp_seq=4 ttl=64 time=1.32 ms
64 bytes from 10.90.10.104: icmp_seq=5 ttl=64 time=1.02 ms

--- 10.90.10.104 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 10ms
rtt min/avg/max/mdev = 0.980/1.143/1.321/0.125 ms
luis@kutulu:~$ ping -c 5 10.90.10.100
PING 10.90.10.100 (10.90.10.100) 56(84) bytes of data.
64 bytes from 10.90.10.100: icmp_seq=1 ttl=64 time=0.803 ms
64 bytes from 10.90.10.100: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 10.90.10.100: icmp_seq=3 ttl=64 time=1.02 ms
64 bytes from 10.90.10.100: icmp_seq=4 ttl=64 time=0.997 ms
64 bytes from 10.90.10.100: icmp_seq=5 ttl=64 time=1.01 ms

--- 10.90.10.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 31ms
rtt min/avg/max/mdev = 0.803/0.969/1.024/0.083 ms
luis@kutulu:~$ 
```

Puerto 20

```

Puerto 11
root@Miguel:/home/miguelcordoba# ping 10.90.10.100 -c 3
PING 10.90.10.100 (10.90.10.100) 56(84) bytes of data.
From 10.90.10.104 icmp_seq=1 Destination Host Unreachable
From 10.90.10.104 icmp_seq=2 Destination Host Unreachable
From 10.90.10.104 icmp_seq=3 Destination Host Unreachable
...
--- 10.90.10.100 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2039ms
pipe 3
root@Miguel:/home/miguelcordoba# ping 10.90.10.6 -c 3
PING 10.90.10.6 (10.90.10.6) 56(84) bytes of data.
64 bytes from 10.90.10.6: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 10.90.10.6: icmp_seq=2 ttl=64 time=1.32 ms
64 bytes from 10.90.10.6: icmp_seq=3 ttl=64 time=2.05 ms
...
--- 10.90.10.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.288/1.553/2.051/0.354 ms
root@Miguel:/home/miguelcordoba# █ significa que el tráfico está etiquetado con esa ID de VLAN

```

```

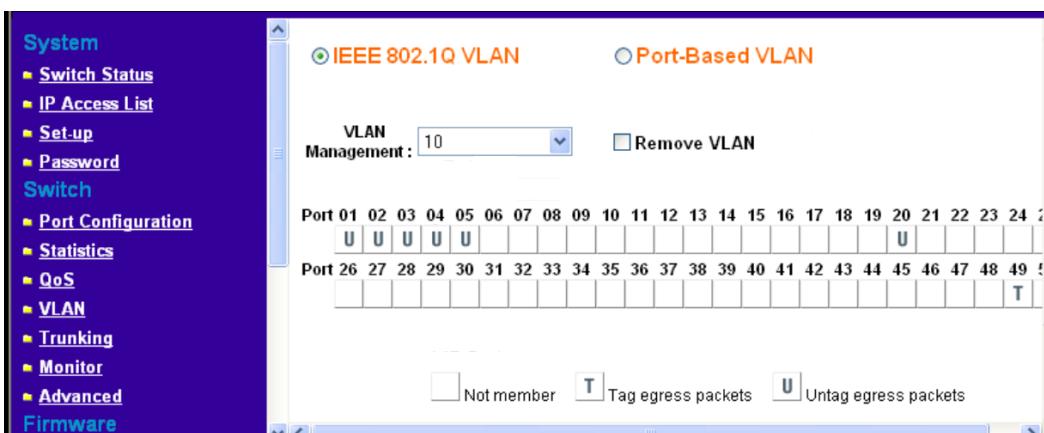
Puerto 1
alexr@pc-alex:~$ ping 10.90.10.104 -c 5
PING 10.90.10.104 (10.90.10.104) 56(84) bytes of data.
From 10.90.10.100 icmp_seq=1 Destination Host Unreachable
From 10.90.10.100 icmp_seq=2 Destination Host Unreachable
From 10.90.10.100 icmp_seq=3 Destination Host Unreachable
From 10.90.10.100 icmp_seq=4 Destination Host Unreachable
From 10.90.10.100 icmp_seq=5 Destination Host Unreachable
...
--- 10.90.10.104 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4098ms
pipe 4
alexr@pc-alex:~$ ping 10.90.10.6 -c 5
PING 10.90.10.6 (10.90.10.6) 56(84) bytes of data.
64 bytes from 10.90.10.6: icmp_seq=1 ttl=64 time=1.06 ms
64 bytes from 10.90.10.6: icmp_seq=2 ttl=64 time=1.27 ms
64 bytes from 10.90.10.6: icmp_seq=3 ttl=64 time=1.30 ms
64 bytes from 10.90.10.6: icmp_seq=4 ttl=64 time=1.30 ms
64 bytes from 10.90.10.6: icmp_seq=5 ttl=64 time=1.30 ms
...
--- 10.90.10.6 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 1.068/1.252/1.308/0.103 ms
alexr@pc-alex:~$ █

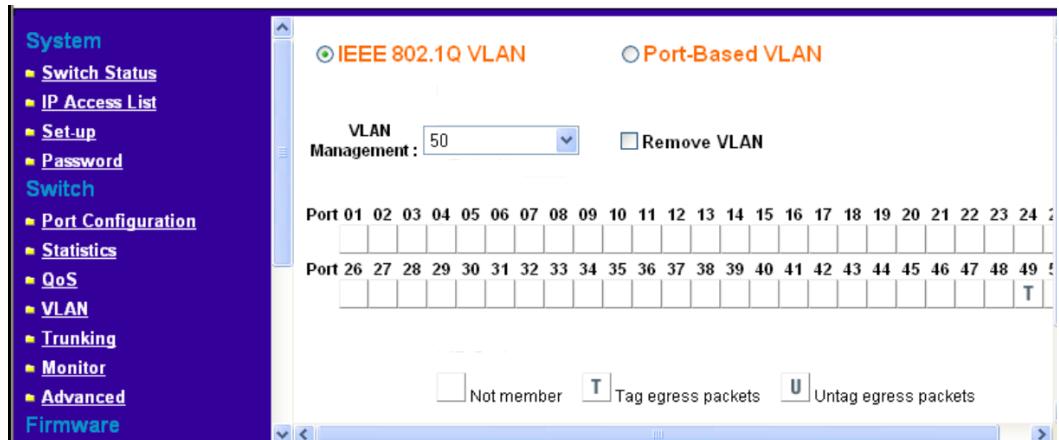
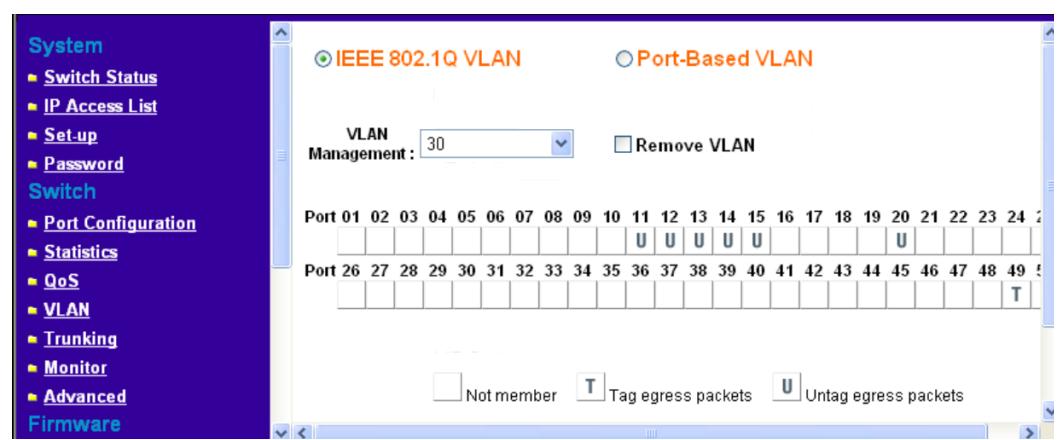
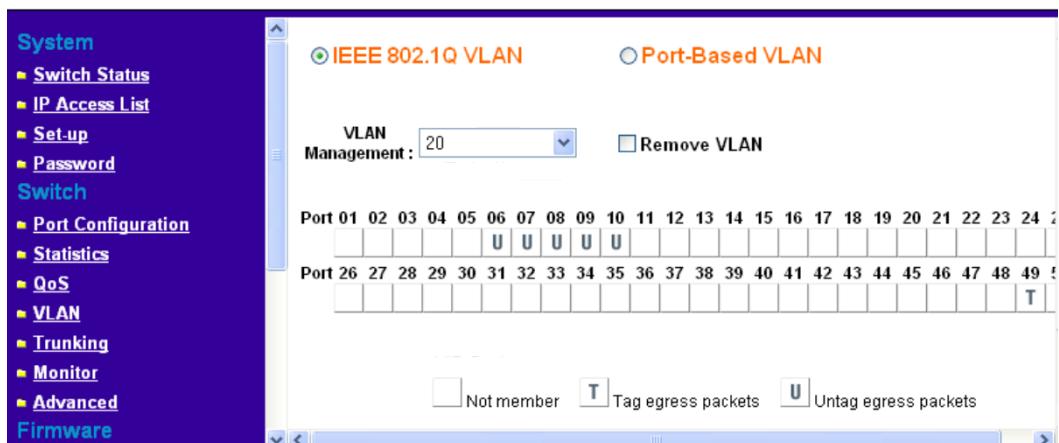
```

7 Enlace entre VLANs

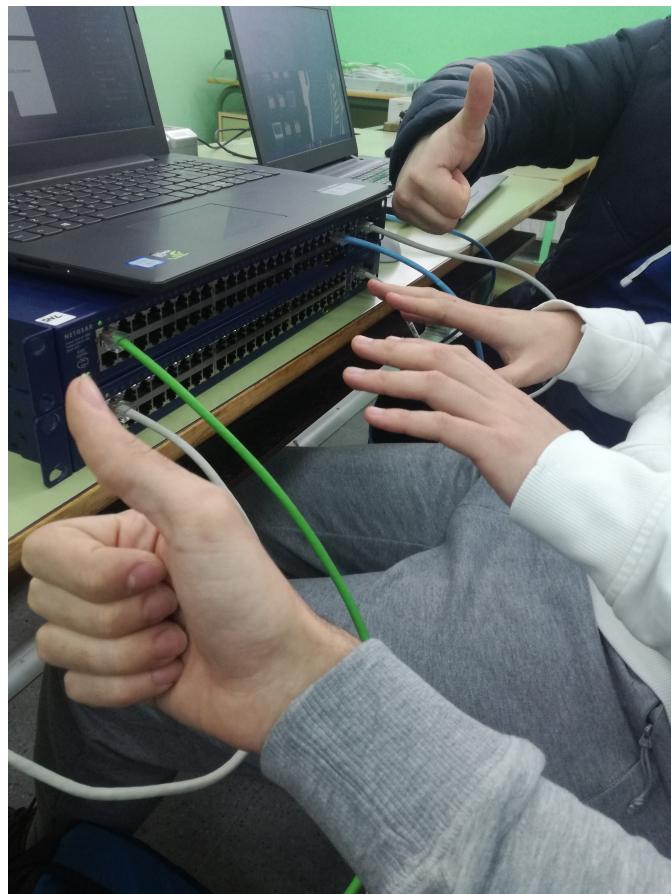
Configuramos ambos Switches con las mismas VLANs del Punto 6.

Añadimos el Puerto 49 en modo Tagged en todas las VLAN de ambos Switches.





Nos conectamos a ambos Switch en la misma VLAN configurada en ambos.



Confirmamos que efectivamente al conectarlos entre dos Switch en la misma VLAN, nos podemos hacer ping.

```
luis@kutulu:~$ ping -c 5 10.90.10.100
PING 10.90.10.100 (10.90.10.100) 56(84) bytes of data.
64 bytes from 10.90.10.100: icmp_seq=1 ttl=64 time=1.65 ms
64 bytes from 10.90.10.100: icmp_seq=2 ttl=64 time=1.02 ms
64 bytes from 10.90.10.100: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 10.90.10.100: icmp_seq=4 ttl=64 time=0.979 ms
64 bytes from 10.90.10.100: icmp_seq=5 ttl=64 time=1.20 ms

--- 10.90.10.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 8ms
rtt min/avg/max/mdev = 0.979/1.171/1.646/0.252 ms
luis@kutulu:~$ ping -c 5 10.90.10.100
PING 10.90.10.100 (10.90.10.100) 56(84) bytes of data.

--- 10.90.10.100 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 108ms
luis@kutulu:~$ 
```

*El ping de abajo se realizó al cambiar a distinta VLAN

8 Equipos en múltiples VLANs

Para poder ponerle una IP a nuestras VLANs en un dispositivo linux primero debemos descargarnos un paquete:

```
apt-get install vlan
```

Y ejecutamos los siguientes comandos:

Empleados →

```
ip link set {Interfaz de red} up  
vconfig add {Interfaz de red} 10  
ip link set {Interfaz de red}.10 up
```

Estudiantes →

```
ip link set {Interfaz de red} up  
vconfig add {Interfaz de red} 20  
ip link set {Interfaz de red}.20 up
```

Administración →

```
ip link set {Interfaz de red} up  
vconfig add {Interfaz de red} 30  
ip link set {Interfaz de red}.30 up
```

Para cambiar de IP solo tendremos que elegir la interfaz que se nos ha creado:

```
ip a add {IP} dev {Interfaz de red}
```

- Empleados: 192.168.10.0/24
- Estudiantes: 192.168.20.0/24
- Administración: 192.168.30.0/24

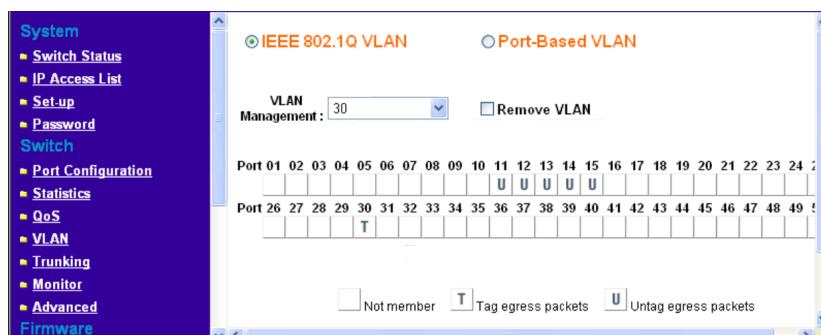
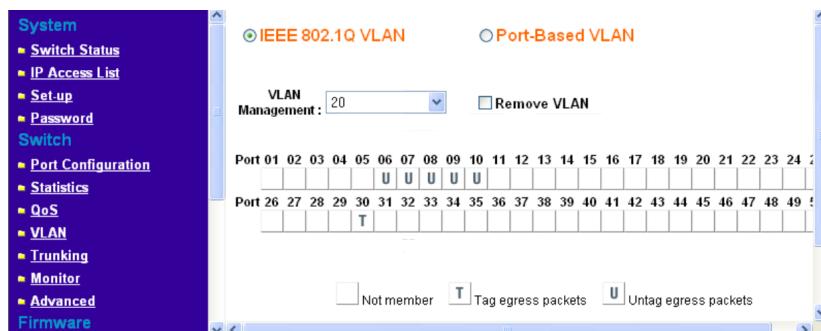
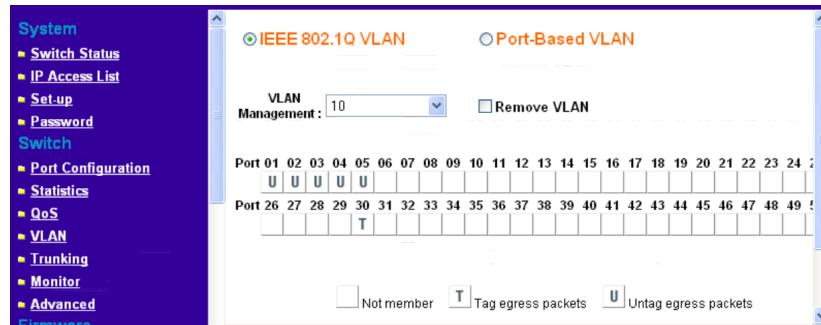
```

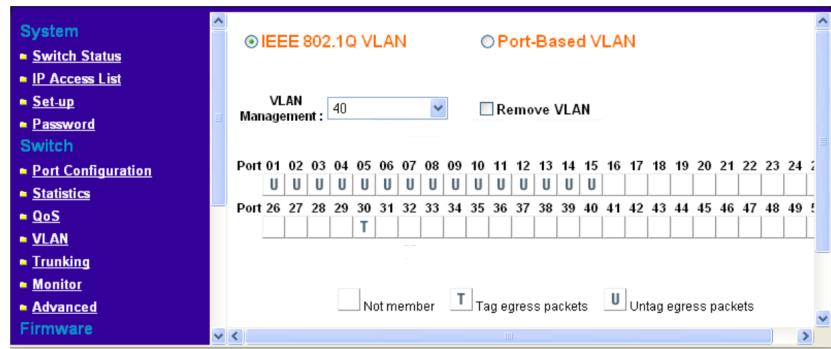
valid lft forever preferred_lft forever
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group default qlen 1000
link/ether 52:54:00:7b:d0:ce brd ff:ff:ff:ff:ff:ff
11: eth0.10@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
link/ether c8:5b:76:eb:69:97 brd ff:ff:ff:ff:ff:ff
inet 192.168.10.0/24 scope global eth0.10
    valid_lft forever preferred_lft forever
    inet6 fe80::ca5b:76ff:feeb:6997/64 scope link
        valid_lft forever preferred_lft forever
12: eth0.20@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
link/ether c8:5b:76:eb:69:97 brd ff:ff:ff:ff:ff:ff
inet 192.168.20.0/24 scope global eth0.20
    valid_lft forever preferred_lft forever
    inet6 fe80::ca5b:76ff:feeb:6997/64 scope link
        valid_lft forever preferred_lft forever
13: eth0.30@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
link/ether c8:5b:76:eb:69:97 brd ff:ff:ff:ff:ff:ff
inet 192.168.30.0/24 scope global eth0.30
    valid_lft forever preferred_lft forever
    inet6 fe80::ca5b:76ff:feeb:6997/64 scope link
        valid_lft forever preferred_lft forever
root@Miguel:/home/miguelcordoba# 

```

IMPORTANTE:

Poner tagged el Puerto 30 que es el que va a escuchar.





Y comprobamos que el Puerto 30 en este caso, consigue escuchar a todas las VLANs.

```
root@Miguel:/home/miguelcordoba# tcpdump -i eth0.10
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0.10, link-type EN10MB (Ethernet), capture size 262144 bytes
12:29:38.732873 IP 192.168.10.6 > Miguel: ICMP echo request, id 23686, seq 1, length 64
12:29:38.732941 IP Miguel > 192.168.10.6: ICMP echo reply, id 23686, seq 1, length 64
12:29:39.734845 IP 192.168.10.6 > Miguel: ICMP echo request, id 23686, seq 2, length 64
12:29:39.734919 IP Miguel > 192.168.10.6: ICMP echo reply, id 23686, seq 2, length 64
12:29:40.736591 IP 192.168.10.6 > Miguel: ICMP echo request, id 23686, seq 3, length 64
12:29:40.736660 IP Miguel > 192.168.10.6: ICMP echo reply, id 23686, seq 3, length 64
12:29:41.738297 IP 192.168.10.6 > Miguel: ICMP echo request, id 23686, seq 4, length 64
12:29:41.738353 IP Miguel > 192.168.10.6: ICMP echo reply, id 23686, seq 4, length 64
12:29:42.740003 IP 192.168.10.6 > Miguel: ICMP echo request, id 23686, seq 5, length 64
12:29:42.740073 IP Miguel > 192.168.10.6: ICMP echo reply, id 23686, seq 5, length 64
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

```
root@Miguel:/home/miguelcordoba# tcpdump -i eth0.20
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0.20, link-type EN10MB (Ethernet), capture size 262144 bytes
12:30:25.780568 IP 192.168.20.1 > Miguel: ICMP echo request, id 12670, seq 2, length 64
12:30:25.780642 IP Miguel > 192.168.20.1: ICMP echo reply, id 12670, seq 2, length 64
12:30:26.804606 IP 192.168.20.1 > Miguel: ICMP echo request, id 12670, seq 3, length 64
12:30:26.804685 IP Miguel > 192.168.20.1: ICMP echo reply, id 12670, seq 3, length 64
12:30:27.828502 IP 192.168.20.1 > Miguel: ICMP echo request, id 12670, seq 4, length 64
12:30:27.828575 IP Miguel > 192.168.20.1: ICMP echo reply, id 12670, seq 4, length 64
12:30:28.852611 IP 192.168.20.1 > Miguel: ICMP echo request, id 12670, seq 5, length 64
12:30:28.852684 IP Miguel > 192.168.20.1: ICMP echo reply, id 12670, seq 5, length 64
12:30:29.959962 ARP, Request who-has 192.168.20.1 tell Miguel, length 28
12:30:29.960200 ARP, Reply 192.168.20.1 is-at 8c:16:45:ea:a4:a0 (oui Unknown), length 46
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

```
root@Miguel:/home/miguelcordoba# tcpdump -i eth0.30
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0.30, link-type EN10MB (Ethernet), capture size 262144 bytes
12:38:39.734991 IP Miguel.mdns > 224.0.0.251.mdns: 0*- [0q] 2/0/0 (Cache flush) PTR Miguel.local., (Cache flush
12:38:42.403435 ARP, Request who-has Miguel tell 192.168.30.1, length 46
12:38:42.403498 ARP, Reply Miguel is-at c8:5b:76:eb:69:97 (oui Unknown), length 28
12:38:42.404390 IP 192.168.30.1 > Miguel: ICMP echo request, id 391, seq 1, length 64
12:38:42.404438 IP Miguel > 192.168.30.1: ICMP echo reply, id 391, seq 1, length 64
12:38:42.496032 IP Miguel.mdns > 224.0.0.251.mdns: 0 [7q] PTR (QM)? nfs._tcp.local. PTR (QM)? _ftp._tcp.local.
(QM)? sftp-ssh._tcp.local. PTR (QM)? smb._tcp.local. PTR (QM)? afpovertcp._tcp.local. (118)
12:38:43.405321 IP 192.168.30.1 > Miguel: ICMP echo request, id 391, seq 2, length 64
12:38:43.405389 IP Miguel > 192.168.30.1: ICMP echo reply, id 391, seq 2, length 64
12:38:44.407007 IP 192.168.30.1 > Miguel: ICMP echo request, id 391, seq 3, length 64
12:38:44.407070 IP Miguel > 192.168.30.1: ICMP echo reply, id 391, seq 3, length 64
12:38:45.408759 IP 192.168.30.1 > Miguel: ICMP echo request, id 391, seq 4, length 64
12:38:45.408823 IP Miguel > 192.168.30.1: ICMP echo reply, id 391, seq 4, length 64
12:38:46.409422 IP 192.168.30.1 > Miguel: ICMP echo request, id 391, seq 5, length 64
12:38:46.409490 IP Miguel > 192.168.30.1: ICMP echo reply, id 391, seq 5, length 64
```

9 Conclusión

Hemos aprendido a configurar las VLAN de un Switch siendo de distintos métodos.