

Enrutamento Con Debian Server mediante ficheiros de configuración

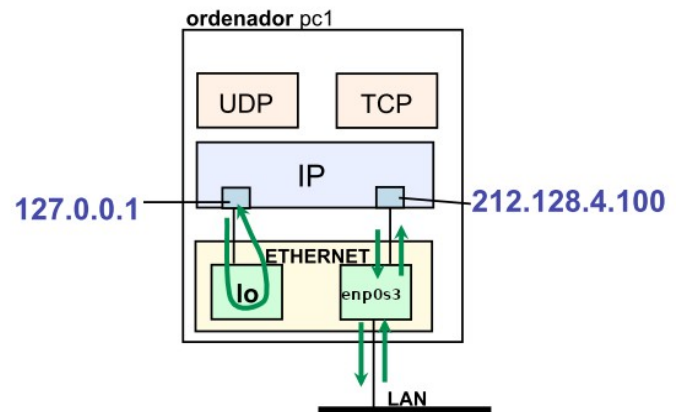
Para as prácticas de enrutamento e configuración IPv4, imos traballar con máquinas virtuais con Debian Server 11.5, sen interfaz gráfico. Isto nos permitirá traballar con pouca RAM (500MB deben ser suficientes), e practicar coa configuración de rede mediante ficheiros de configuración.

Interfaces de rede dunha máquina Debian Server

Todas as máquinas teñen sempre a interfaz de rede **lo** (interfaz de loopback), que é unha interfaz de autoenvío.

Unha máquina Debian Server cunha tarxeta Ethernet terá, ademais da interfaz **lo**, unha interfaz **enp0s3**, que terá que ser configurada.

Cando se engaden máis tarxetas debemos comprobar o seu nome, para proceder á súa configuración.



Consulta de configuración

ip a l – (*ip address list*) ou ***ip a s*** (*ip adress show*) ou ***ip a***

Para obter a configuración de rede dun equipo empregaremos o comando ***ip a l*** (*ip address list*) o simplemente ***ip a*** na terminal. Por exemplo se temos a máquina virtual (MV) coa configuración de Virtual Box en NAT obteremos:

```
redes@DebianServer:~$ ip a l
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
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:01:07:9a brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86400sec preferred_lft 86400sec
    inet6 fe80::a00:27ff:fe01:79a/64 scope link
        valid_lft forever preferred_lft forever
redes@DebianServer:~$
```

Configuración permanente. Ficheiros de configuración

Nas distribución linux Debian e derivados, a configuración de rede está no ficheiro `/etc/network/interfaces`. Editaremos o ficheiro con

`sudo nano /etc/network/interfaces`

O ficheiro `/etc/network/interfaces` estará no inicio así:

```
redes@DebianServer:~$ more /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

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp
```

As primeiras liñas corresponden co bucle local:

```
auto lo
iface lo inet loopback
```

e despois está configurada a tarxeta `enp0s3`, para que colla a configuración por DHCP:

```
allow-hotplug enp0s3
iface enp0s3 inet dhcp
```

Para asignar unha IP teremos que modificar as liñas que corresponden á tarxeta `enp0s3` decíndolle que empregaremos direccionamento estático (coa palabra `static`), e asignamos a IP `192.168.1.2`, con máscara `/24`:

```
auto enp0s3
iface enp0s3 inet static
address 192.168.1.2
netmask 255.255.255.0
```

O ficheiro completo debe quedar así (non se pode eliminar o bucle local!) :

```
# 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

# The primary network interface
auto enp0s3
iface enp0s3 inet static
address 192.168.1.2
netmask 255.255.255.0
```

Hai que ser coidadoso en todos os caracteres, como na dobre 'd' de `address`, para que non haxa erros.

Unha vez realizados os cambios no ficheiro `/etc/network/interfaces`, executaremos **calquera** das seguintes ordes:

```
sudo /etc/init.d/networking restart 0U
sudo systemctl restart networking.service
```

e comprobaremos que a nova IP está en uso con **ip a**:

```
redes@DebianServer:~$ ip a
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
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
    link/ether 08:00:27:01:07:9a brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.2/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe01:79a/64 scope link tentative
        valid_lft forever preferred_lft forever
redes@DebianServer:~$
```

Configuración da porta de enlace

Para completar a configuración de rede, ademais da IP e a máscara de subrede, temos tamén que configurar unha porta de enlace (router, enrutador, encamiñador ou gateway) que leve os paquetes fóra da nosa rede, e un ou dous servidores DNS.

Para configurar a porta de enlace (gateway) engadimos a liña coa IP da porta de enlace:

```
gateway 192.168.1.1
```

de modo que o noso ficheiro *etc/network/interfaces* quedará:

```
redes@DebianServer:~$ more /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

# The primary network interface
auto enp0s3
iface enp0s3 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
```

Se tivéssemos conexión a internet, deberíamos tamén configurar o servidor DNS (neste exemplo o servidor DNS de google), para poder navegar, engadindo a liña:

```
dns-nameservers 8.8.8.8
```

O ficheiro de configuración de rede completo quedaría así:

```
# 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

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.1.2
    netmask 255.255.255.0
    gateway 192.168.1.1
    dns-nameservers 8.8.8.8
```

Configuración de rede manual

Aínda que non será o xeito de traballar nas prácticas, en ocasións pode ser interesante configurar unha interface e unha porta de enlace manualmente, sen modificalo de xeito permanente. Para isto empregamos o comando **ip**. Para configurar unha IP e máscara temos primeiro que eliminar a IP previa, para non incorporar a nova como segunda do interface. Teremos pois que facer:

```
sudo ip addr del 192.168.1.2/24 dev enp0s3
```

Engadimos agora a nova IP e máscara:

```
sudo ip addr add 192.168.200.2/24 dev enp0s3
```

Para incorporar un novo gateway, borramos o anterior, e engadimos o novo:

```
sudo ip del default via 192.168.1.1
sudo ip addr default via 192.168.200.1
```

Apagar a MV Debian

Para apagar un sistema Linux desde liña de comandos, podemos empregar calquera dos seguintes comandos:

```
sudo shutdown -h 0
```

```
sudo halt -p
```

```
sudo poweroff
```

```
sudo shutdown -h -P 0
```

Configuración de Debian Server como Router

Se queremos que unha máquina virtual Debian Server faga de router, terá que ter varias tarxetas de rede configuradas (unha para cada subrede á que estea conectado) , e habilitar o enrutamento.

Para habilitar o enrutamento, temos que editar o ficheiro */etc/sysctl.conf*:

```
sudo nano /etc/sysctl.conf
```

Temos que “descomentar” ou quitar o # antes da liña do enrutamento IPv4

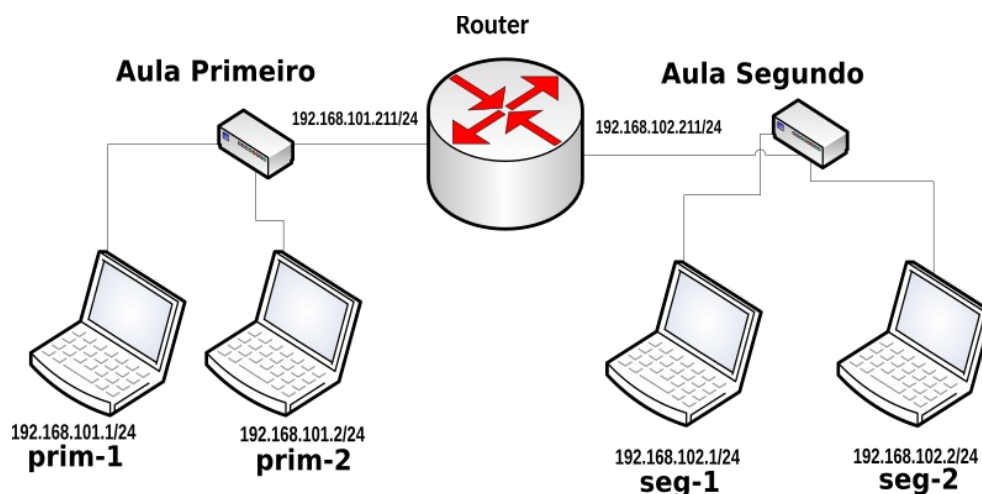
```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward = 1
```

E reiniciar o equipo, ou executar

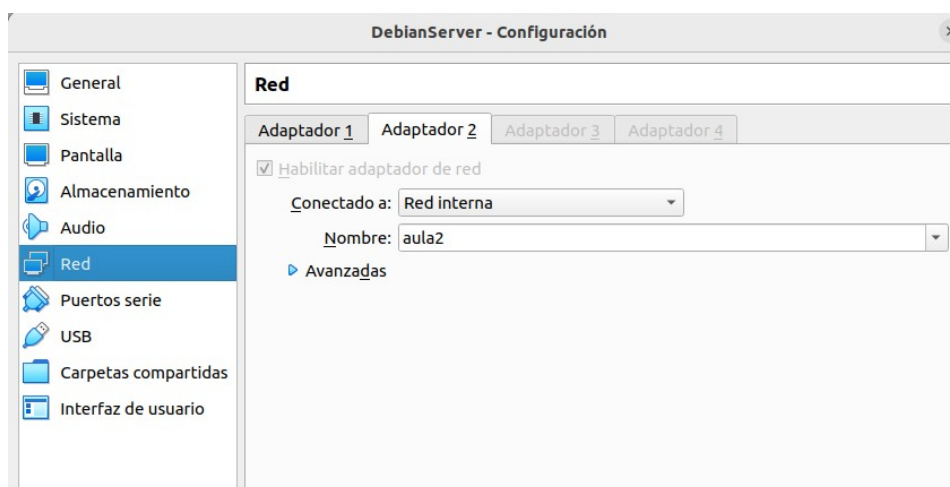
```
sudo sysctl -p
```

Se temos varios routers teremos que definir as rutas, como se verá máis adiante.

Con esta configuración a máquina virtual (MV) fará de router entre 2 subredes coas que estea conectado directamente, como no seguinte exemplo:



Para isto tería que ter configuradas as 2 tarxetas de rede no VirtualBox, previamente:



E configurar o ficheiro para ter as 2 tarxetas, en redes distintas:

```
# 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

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.101.211
    netmask 255.255.255.0

auto enp0s8
iface enp0s8 inet static
    address 192.168.102.211
    netmask 255.255.255.0
```

Cun escenario como o mostrado neste exemplo, debería ser posible xa facer ping desde a 192.168.101.1 a 192.168.102.1, sempre que cada un teña configurado como gateway a IP da tarxeta de rede do router que corresponde:

prim-1

```
iface enp0s3 inet static
    address 192.168.101.1
    netmask 255.255.255.0
    gateway 192.168.101.211
```

seg-1

```
iface enp0s3 inet static
    address 192.168.102.1
    netmask 255.255.255.0
    gateway 192.168.102.211
```

Mostrar o tráfico que pasa por unha tarxeta de rede

Empregaremos o comando

```
tcpdump -i enp0s3
```

Deste xeito todos os paquetes recibidos mostraranse na terminal. Se tcmdump non está instalado na máquina virtual haberá que instalalo previamente:

```
sudo apt install tcpdump
```

OLLO: Para poder instalar **tcpdump** deberás ter acceso a internet, por exemplo configurar a rede da máquina en modo ponte na rede do teu equipo real, co gateway a IP do teu router.

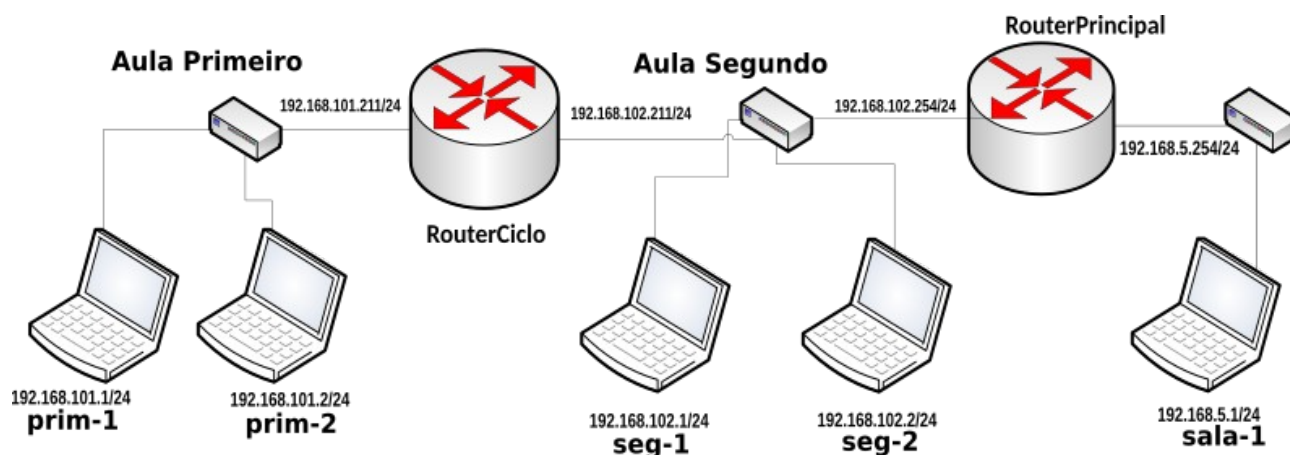
Tamén podes baixar a MV co tcpdump instalado neste enlace:

<https://drive.google.com/file/d/18x9SbgjZC5nPKI5TEZRyh3YTSryPZ-XC/view?usp=sharing>

Escenario con 2 routers

No escenario anterior, o router estaba conectado directamente á rede destino, polo que sabía que o destino estaba na rede dalgunha das súas tarxetas. Isto cambia cando engadimos un router a maiores, pois o router non coñece a onde ten que enviar un paquete: temos que definir as rutas.

Supoñamos o seguinte escenario:



Se un ordenador da Aula de Primeiro quere enviar un paquete a sala-1, enviarálo ao seu gateway (RouterCiclo), pero este non sabe que facer con el, pois non o pode entregar directamente. Para iso teríamos que incorporar na súa táboa de enrutamento a regra para alcanzar esa subrede. Podemos facer lo desde a liña de comando con ip route:

RouterCiclo:

```
sudo ip route add 192.168.5.0/24 via 192.168.102.254
```

RouterPrincipal

```
sudo ip route add 192.168.101.0/24 via 192.168.102.211
```

Empregamos a IP da porta de enlace, despois da palabra reservada **via**.

Podemos ver as rutas existentes no sistema con: (son equivalentes)

```
sudo ip route show
sudo ip route list
```

Para **BORRAR** unha ruta poderíamos facer

```
sudo ip route delete 192.168.5.0/24 via 192.168.102.254
```

Como o escenario é pechado, poderíamos indicar que esa ruta é unha ruta xenérica, de este modo:

RouterCiclo:

```
sudo ip route add default via 192.168.102.254
```

RouterPrincipal

```
sudo ip route add default via 192.168.102.211
```

Todos os paquetes que non se entregasen directamente (non están en ningunha das súas subredes) serán entregados cara o outro router.

Se preferimos facelo con ficheiro de configuración teremos que engadir a liña da ruta no `/etc/network/interfaces`:

RouterCiclo:

```
# 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

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.101.211
    netmask 255.255.255.0

auto enp0s8
iface enp0s8 inet static
    address 192.168.102.211
    netmask 255.255.255.0

#RUTAS
up ip route add default via 192.168.102.254
```

Fíxate que a IP que se engade na ruta é a IP á **que se entregarán os paquetes**, no caso de RouterCiclo a IP de RouterPrincipal e viceversa.

RouterPrincipal:

```
# 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

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.102.254
    netmask 255.255.255.0

auto enp0s8
iface enp0s8 inet static
    address 192.168.5.254
    netmask 255.255.255.0

#RUTAS
up ip route add default via 192.168.102.211_
```


OLLO: Cando definimos unha ruta nun router é preciso comprobar:

- A IP despois de “via” **está na rede** dalgunha das tarxetas do router
- A IP despois de “via” **é a IP do seguinte salto**, a IP do próximo router

Comprobando as rutas

Para comprobar as rutas nun equipo podemos empregar o comando **ip route**.

En **routerCiclo**:

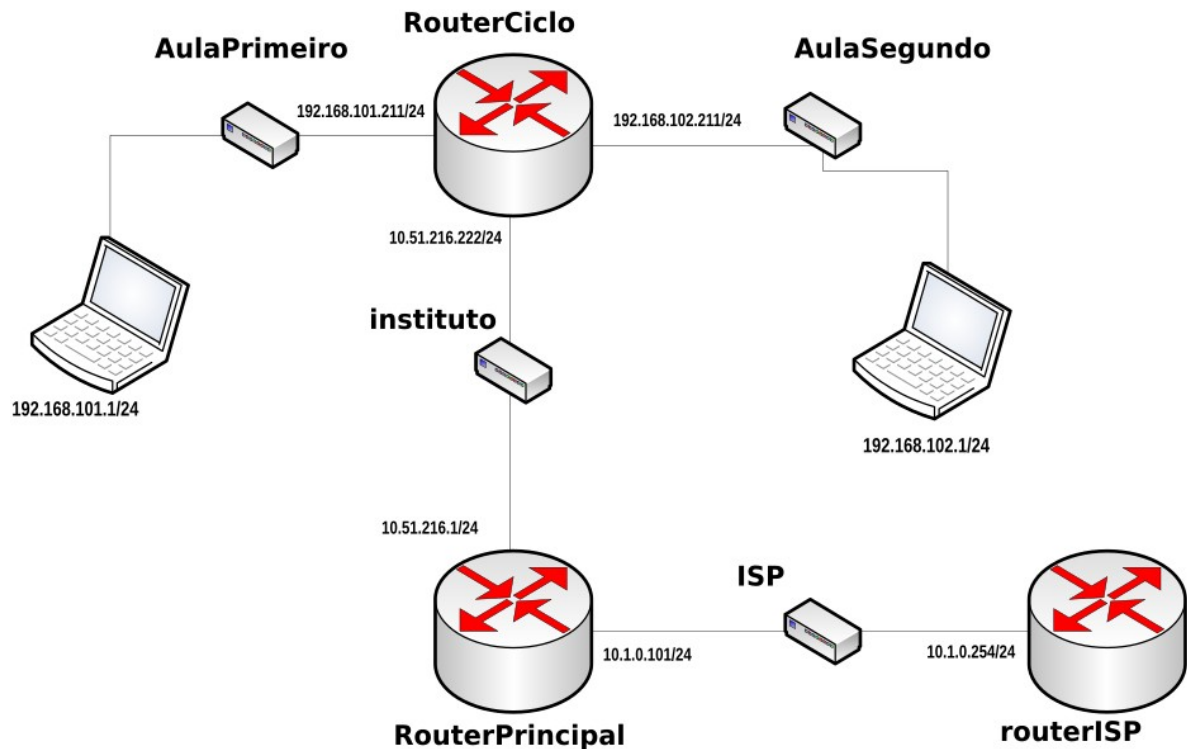
```
redes@routerCiclo:~$ ip route
default via 192.168.102.254 dev enp0s8
192.168.101.0/24 dev enp0s3 proto kernel scope link src 192.168.101.211
192.168.102.0/24 dev enp0s8 proto kernel scope link src 192.168.102.211
```

En **routerPrincipal**:

```
redes@routerPrincipal:~$ ip route
default via 192.168.102.211 dev enp0s3
192.168.5.0/24 dev enp0s8 proto kernel scope link src 192.168.5.254
192.168.102.0/24 dev enp0s3 proto kernel scope link src 192.168.102.254
```

Escenario con 3 routers

No escenario anterior realmente faríanos falta outro switch (que foi omitido por simplicidade) para referir un escenario real. A situación real nun instituto por exemplo cun único ciclo de ASIR podería ser como se amosa no esquema seguinte:



Temos un router de Ciclo (no que poderíamos ter instalado un proxy para controlar o tráfico), e este router está conectado a un switch central do instituto, ao que irían todas as conexións do instituto. Este switch (ou varios conectados entre si no armario principal do centro) deberá estar conectado co router principal, que se conecta a internet, neste exemplo a un router do ISP.

Agora, en todo o escenario:

- todos os paquetes con destino as redes do noso escenario deben ser entregado correctamente.
- calquera paquete con dirección destino NON pertencente a esas redes irá encamiñada cara o router routerISP.

A configuración dos equipos de AulaPrimeiro e AulaSegundo, non cambiará, tan só encamiñarán polo seu gateway.

O RouterCiclo terá modificada a súa configuración. Temos que cambiar as IP's e engadir a terceira tarxeta:

```
auto lo
iface lo inet loopback
auto enp0s3
iface enp0s3 inet static
```

```
        address 192.168.101.211
        netmask 255.255.255.0

auto enp0s8
iface enp0s8 inet static
    address 192.168.102.211
    netmask 255.255.255.0

auto enp0s9
iface enp0s9 inet static
    address 10.51.216.222
    netmask 255.255.255.0

up ip route add default via 10.51.216.1
```

O routerPrincipal deberá agora ter no enrutamento:

- 2 rutas para as aulas (os paquetes con destino as aulas deben ir cara o RouterCiclo
- o resto (default) ir cada o router ISP.

A configuración do **routerPrincipal** debe quedar así:

```
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
    address 10.51.216.1
    netmask 255.255.255.0

auto enp0s8
iface enp0s8 inet static
    address 10.1.0.101
    netmask 255.255.255.0

up ip route add 192.168.101.0/24 via 10.51.216.222
up ip route add 192.168.102.0/24 via 10.51.216.222
up ip route add default via 10.1.0.254
```

Finalmente, o **routerISP** devolverá os paquetes de volta para as redes coñecidas do noso escenario, enviando os paquetes a 10.1.0.101 (o Router Principal)

```
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
    address 10.1.0.254
    netmask 255.255.255.0

up ip route add 192.168.101.0/24 via 10.1.0.101
```

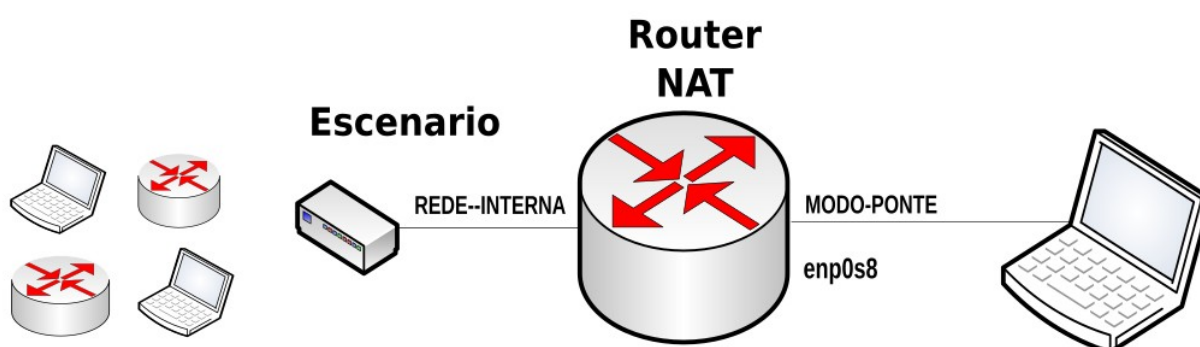
```
up ip route add 192.168.102.0/24 via 10.1.0.101  
up ip route add 10.51.216.0/24 via 10.1.0.101
```

Faltaría neste último router definir unha ruta por defecto, que debería ir cara internet, se queremos dar acceso a internet ao escenario.

Configuración de NAT en Debian Server

Para que o noso escenario teña acceso a internet, deberíamos configurar un router de saída. Se estamos traballando con direccións privadas, este router de saída deberán facer unha conversión entre direccións privadas, e unha dirección pública de saída, o que se coñece como **NAT (Network Address Traslation)**, que veremos polo miúdo posteriormente.

Veremos brevemente como configurar NAT nun Debian Server para dar internet aos nosos escenarios.



O **RouterNAT** deberá ter configurada a enp0s8 en modo PONTE, na rede do Equipo real, e a outra tarxeta enp0s3 na rede interna correspondente. Para que faga NAT por esa tarxeta temos que:

- Engadir a tarxeta enp0s8 desde VirtualBox e configurala en `/etc/network/interfaces`. Neste exemplo o equipo real está na rede 192.168.150.0/24
- Deberá ter habilitado o enrutamento en [/etc/sysctl.conf](#)
- Temos que configurar as nftables :

```
RouterNAT$ sudo nft add table ip nat
RouterNAT$ sudo nft add chain ip nat preenrutado {type nat hook prerouting priority -100 \;}
RouterNAT$ sudo nft add chain ip nat posenrutado {type nat hook postrouting priority 100 \;}
RouterNAT$ sudo nft add rule ip nat posenrutado oifname "enp0s8" masquerade
```

- Podemos ver se engadiron correctamente con:

```
sudo nft list table nat
```

- Despois de sudo su (co rol de root adquirido), executamos

```
RouterNAT# nft list ruleset >> /etc/nftables.conf
```

- Iniciamos o servizo nftables, e o facemos permanente cando se inicia o equipo:

```
RouterNAT# sudo systemctl start nftables.service
RouterNAT# sudo systemctl enable nftables.service
```

Agora desde calquera equipo do noso escenario, se o enrutamento leva cara o ROUTERNAT os paquetes que deben ir a internet, poderíamos acceder a direccións externas.

No caso do escenario de 3 routers, tras preparar o routerISP co anterior teríamos a seguinte configuración (se o gateway do equipo real é a 192.168.150.1/24):

```
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet static
    address 10.1.0.254
    netmask 255.255.255.0

auto enp0s8
iface enp0s8 inet static
    address 192.168.150.254
    netmask 255.255.255.0

up ip route add 192.168.101.0/24 via 10.51.216.222
up ip route add 192.168.102.0/24 via 10.51.216.222
up ip route add 10.51.216.0/24 via 10.51.216.222
up ip route add default via 192.168.150.1
```