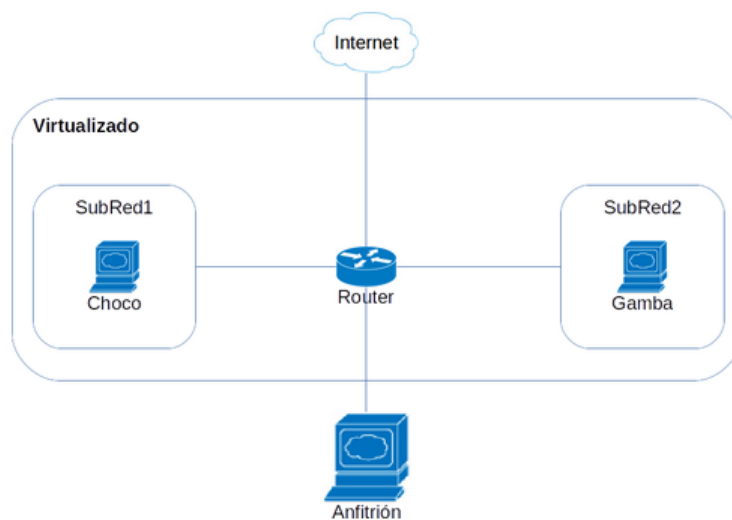


# Memoria Práctica 6

## Administración de Servidores

### Simulación de una infraestructura de red basada en CentOS sobre máquinas virtuales



Almudena García Jurado-Centurión

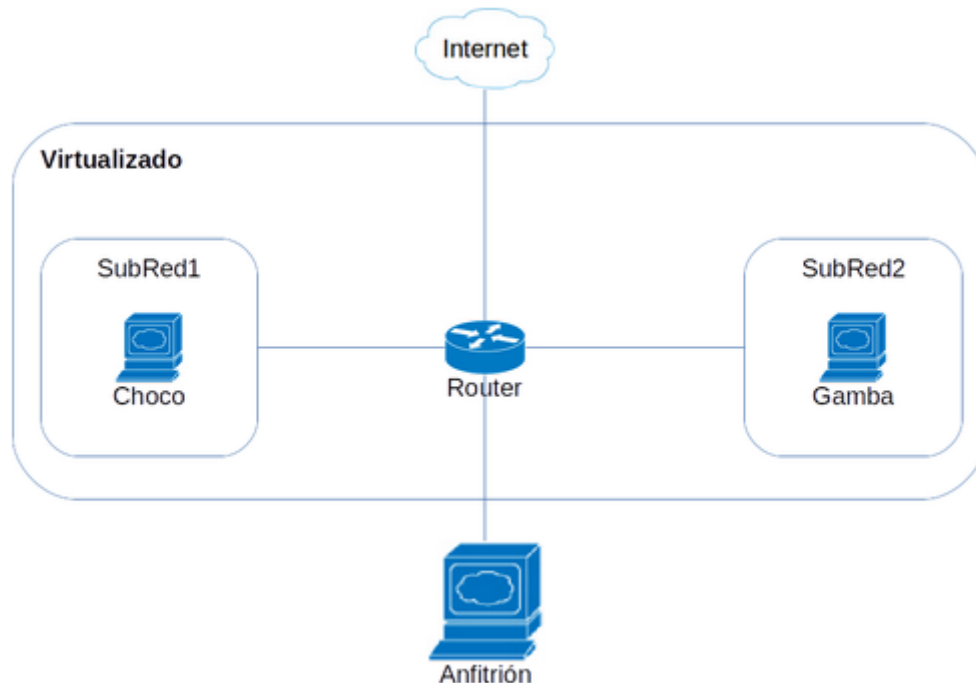
# **Índice**

0. Introducción
1. Creación de las máquinas virtuales
2. Configurando tarjetas de red
  - Router
  - Choco y Gamba
3. Comprobando conectividad en el router
  - eth0
  - eth1
  - eth2 y eth3
4. Activando las interfaces eth0 y eth1 al inicio del sistema
5. Configurando las interfaces eth2 y eth3
  - eth2
  - eth3
6. Conectando Choco al router
7. Conectando Gamba al router
8. Configurando enrutamiento
  - Activando reenvío de paquetes
  - Configurando el NAT en IPTables
9. Asignando IP estática a eth1 en el router
10. Comprobando conectividad en Choco y Gamba
11. Configurando el DNS en Choco y Gamba
12. Resumen

## 0. Introducción

En esta práctica aprenderemos a crear una infraestructura de red mediante máquinas virtuales, y a interconectarlas según un esquema ya conocido.

El esquema a crear es el siguiente:



Para construir la infraestructura, se nos incluyen los siguientes requisitos:

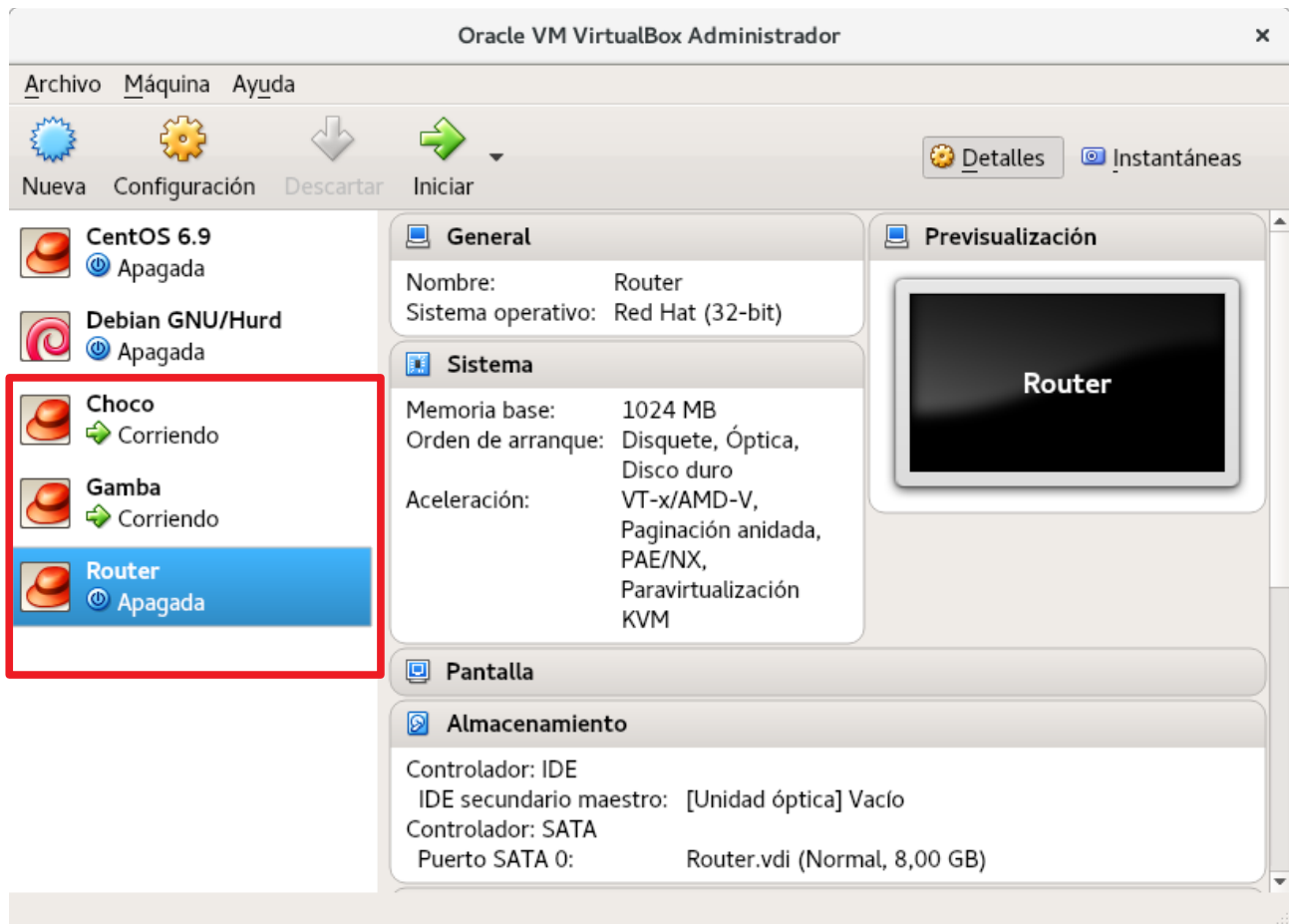
La red estará formada por tres ordenadores:

- Choco: equipado con CentOS, un disco duro y una tarjeta de red.
- Gamba: equipado con CentOS, un disco duro y una tarjeta de red
- Router: equipado con CentOS y cuatro tarjetas de red

El ordenador router actúa como pasarela para las subredes en las que se encuentran gamba y choco. Además, router permite tanto el acceso a internet de todos los equipos (choco, gamba, router) como el acceso del equipo anfitrión al router.

## 1. Creación de las máquinas virtuales

Antes de realizar la interconexión, debemos crear la infraestructura que albergará nuestro esquema de red. Puesto que no tenemos equipos reales, los simularemos mediante máquinas virtuales.



Dado que los requerimientos no son muy altos, únicamente les hemos asignado 1GB de RAM y 8 GB de disco duro a cada uno.

## 2. Configurando tarjetas de red

En VirtualBox, la configuración por defecto para las tarjetas de red es NAT. Esta configuración nos permite acceder a Internet, pero nos impide que el sistema pueda acceder a la máquina virtual.

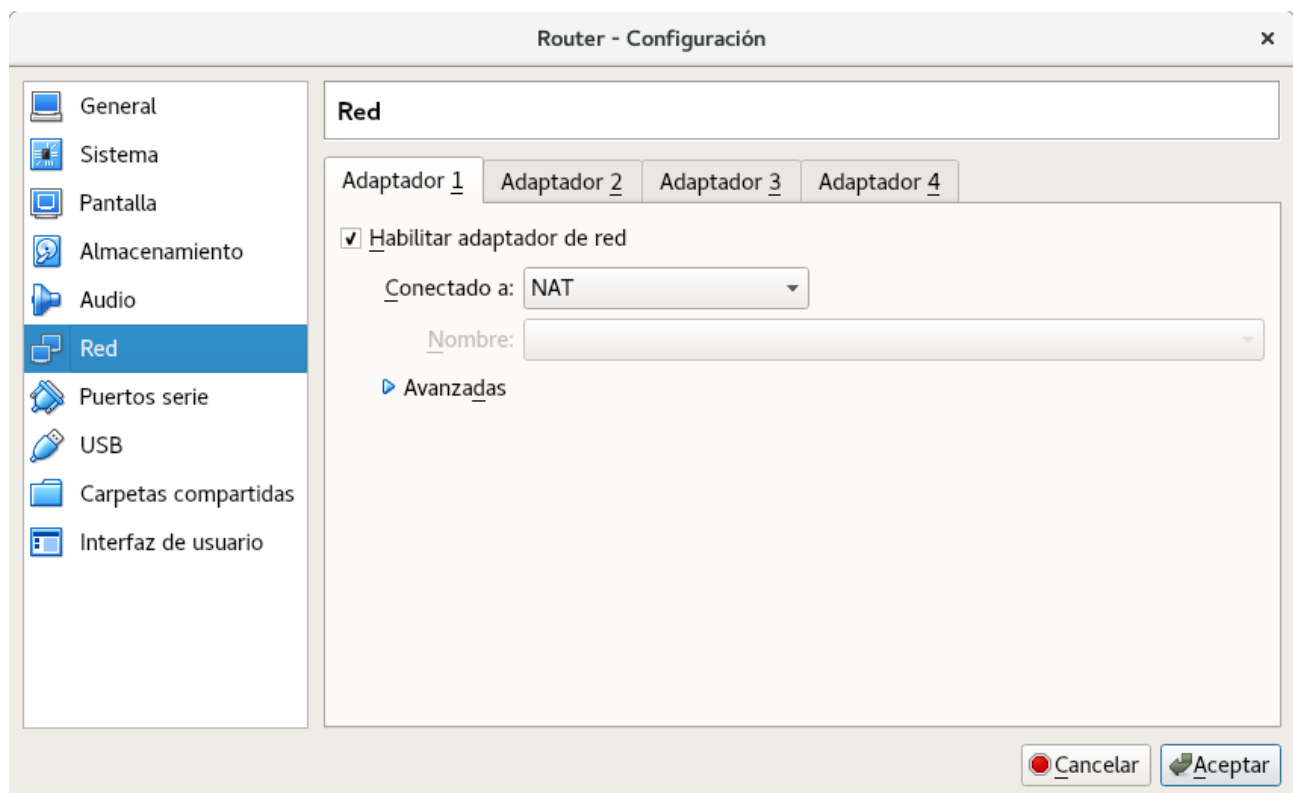
Por esa razón, debemos cambiar la configuración de todas aquellas interfaces que no requieran acceso a Internet, para que puedan interconectarse con otras máquinas virtuales o con el sistema anfitrión.

- **Router**

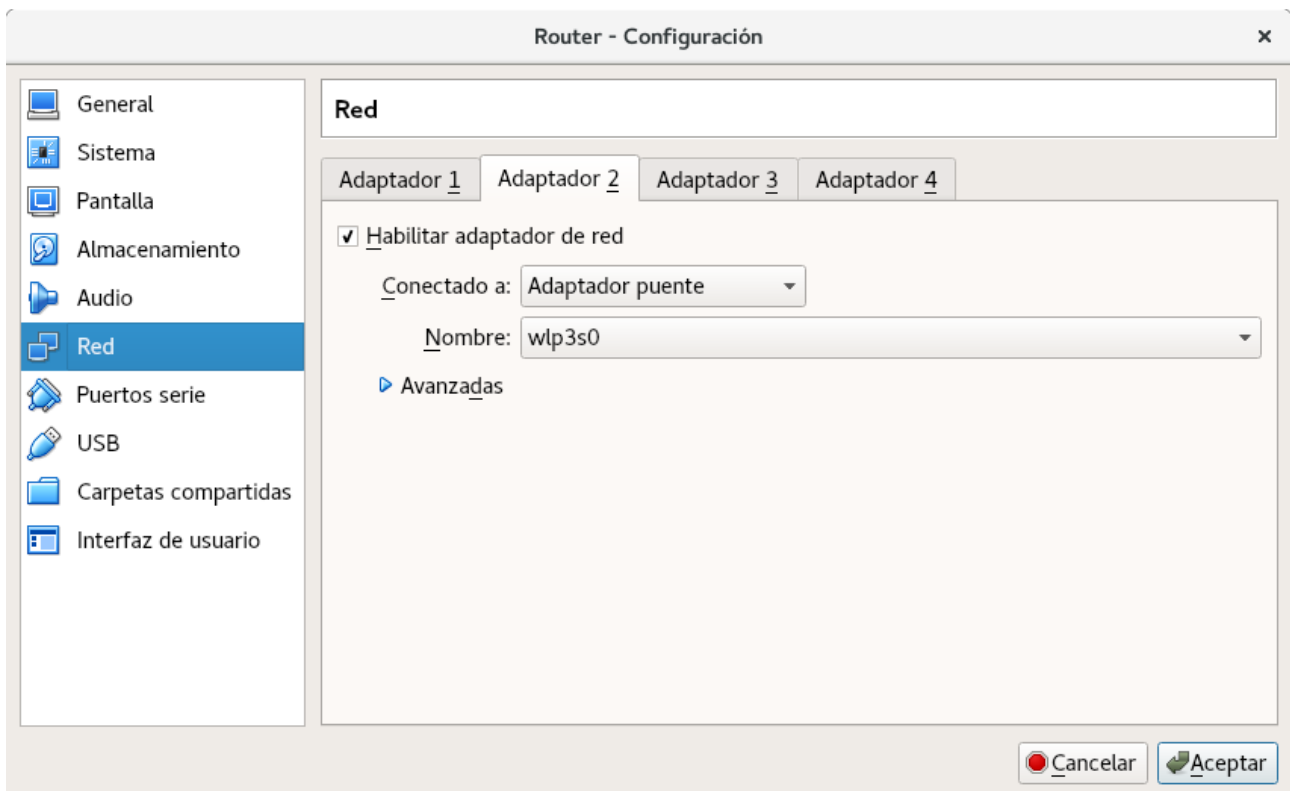
En los requisitos también se nos indica que el router debe tener 4 tarjetas de red, así que las creamos mediante el asistente de VirtualBox

Abrimos la configuración de la máquina virtual y, en la sección de Red, habilitamos los 4 adaptadores de red.

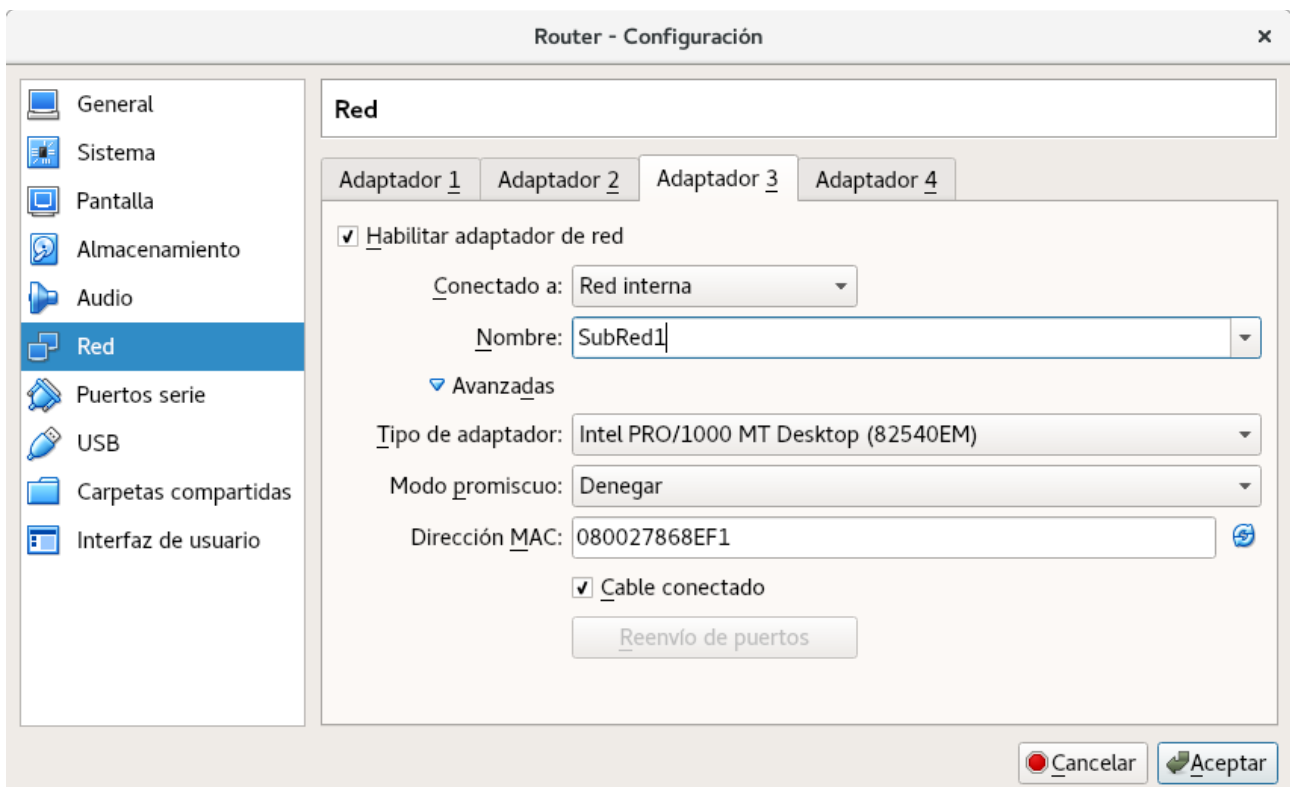
La primera interfaz la usaremos para conectar a Internet, así que la configuraremos como NAT.



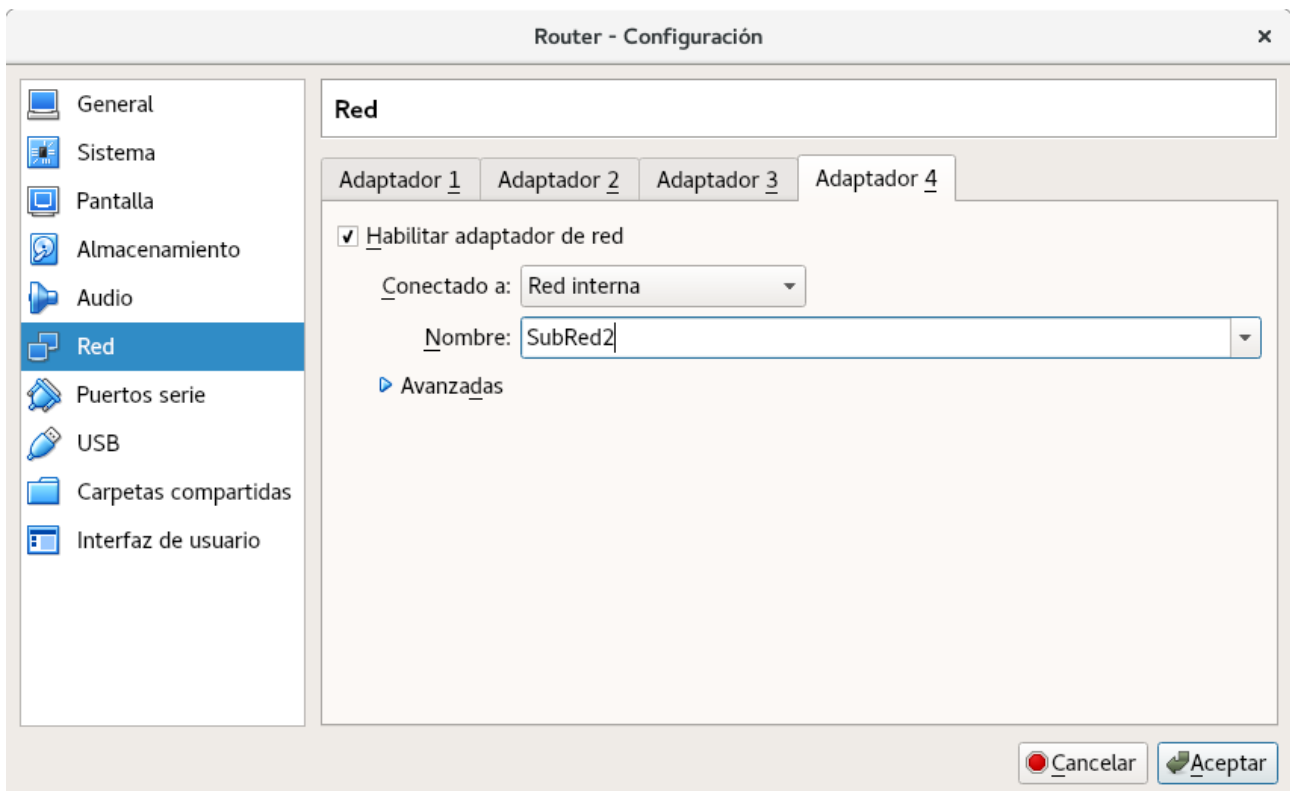
La segunda interfaz la usaremos para conectar con el anfitrión, así que la configuraremos como Adaptador Puente



La tercera interfaz la usaremos para conectar con Choco, así que lo configuraremos como Red Interna, con SubRed1

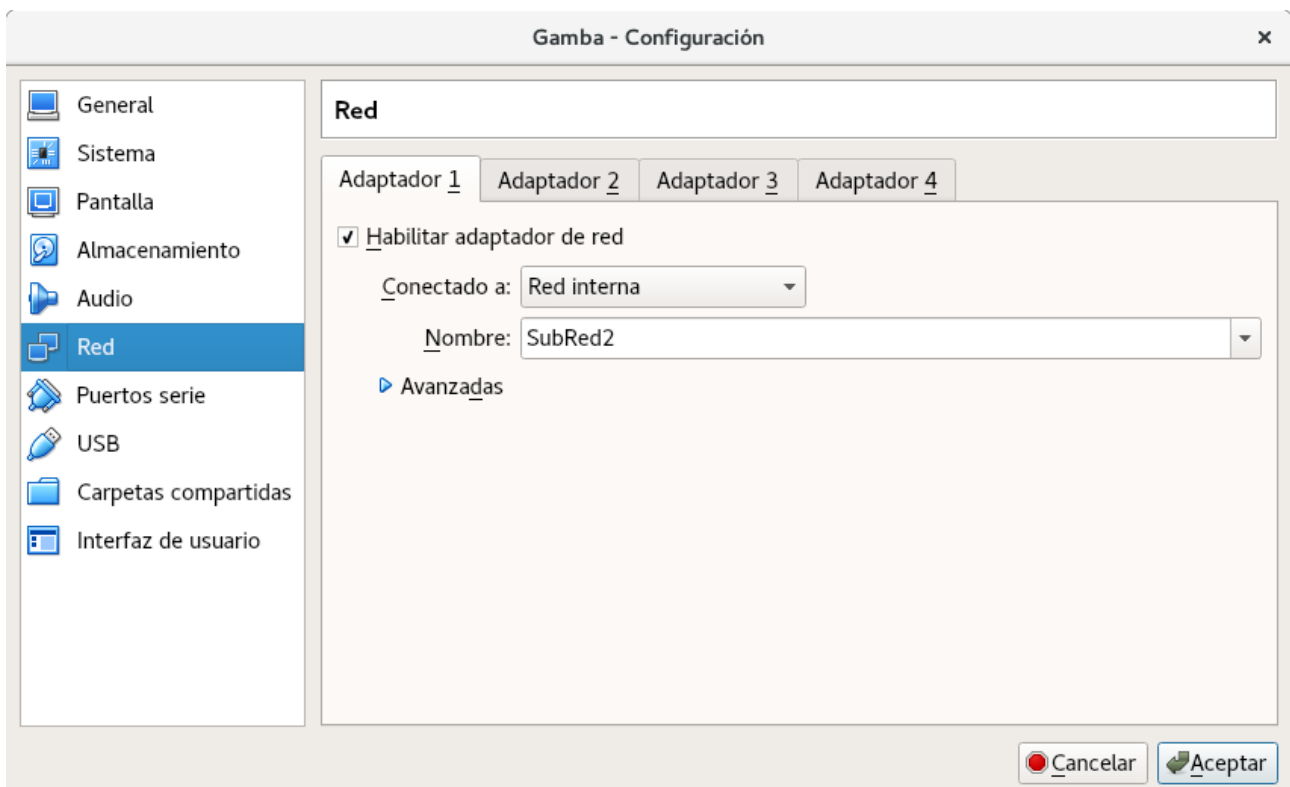
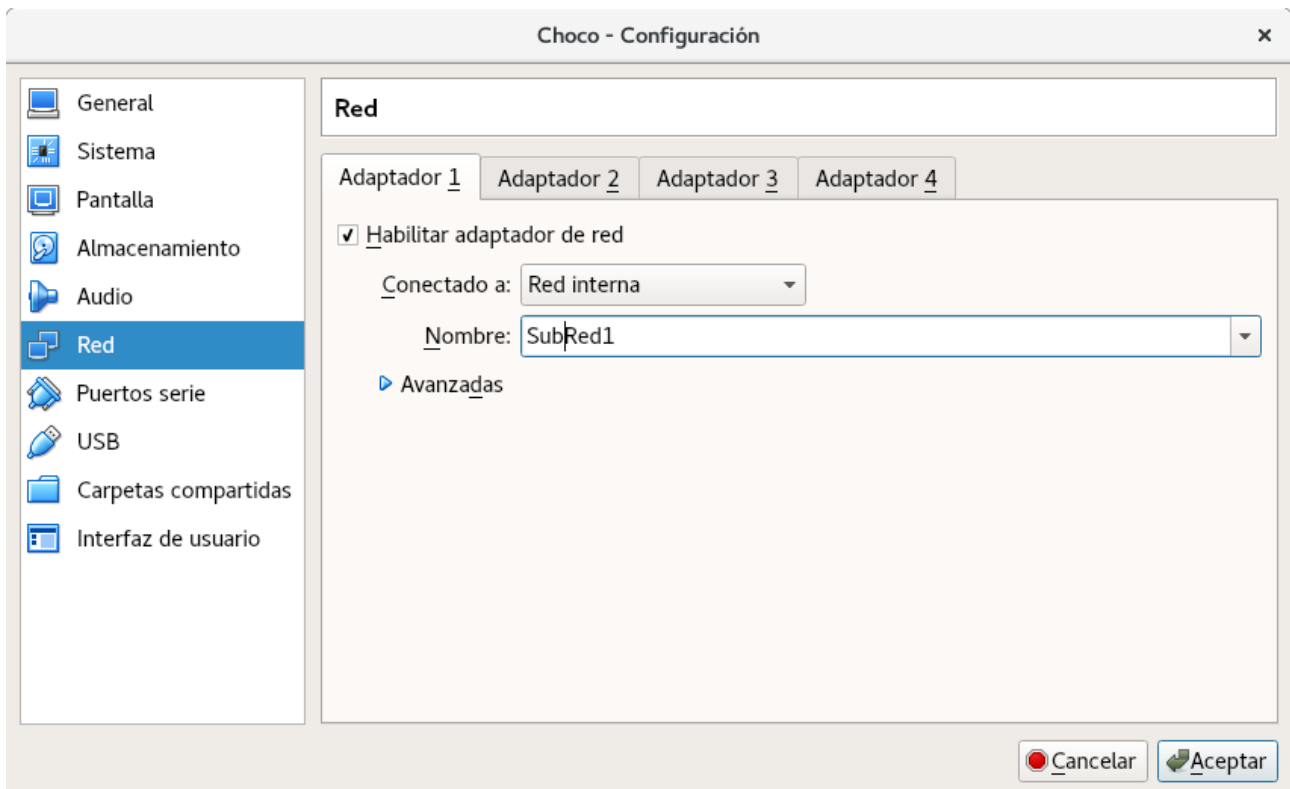


La cuarta interfaz la usaremos para conectar con Gamba, así que la configuraremos como Red Interna, con SubRed2



- **Choco y Gamba**

Configurado el router, continuamos con las máquinas cliente: Choco y Gamba. Estas estarán conectadas a Red Interna, con nombres SubRed1 y SubRed2.





### 3. Comprobando conectividad en el router

Tras configurar las tarjetas de red desde de VirtualBox, comprobamos el estado de las interfaces, y su conectividad.

Lo primero que debemos hacer es comprobar si el sistema detecta las tarjetas de red. Para ello, usaremos el comando `ifconfig`, que nos muestra las interfaces activas en el sistema y las configuraciones asociadas a las mismas (Dirección IP, MAC...)

Ejecutamos el comando y vemos el resultado:

```
[root@router ~]# ifconfig | more
lo                Link encap:Local Loopback
                  inet addr:127.0.0.1  Mask:255.0.0.0
                  inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING  MTU:65536  Metric:1
                  RX packets:4 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:264 (264.0 b)  TX bytes:264 (264.0 b)
```

Vemos que la única interfaz activa es el bucle local (lo), así que ejecutamos `ifconfig -a` para que incluya las interfaces inactivas

```
eth0              Link encap:Ethernet  HWaddr 08:00:27:C8:CC:DC
                  BROADCAST MULTICAST  MTU:1500  Metric:1
                  RX packets:6 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:1534 (1.4 KiB)  TX bytes:1626 (1.5 KiB)

eth1              Link encap:Ethernet  HWaddr 08:00:27:2D:24:D4
                  BROADCAST MULTICAST  MTU:1500  Metric:1
                  RX packets:160 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:12065 (11.7 KiB)  TX bytes:1902 (1.8 KiB)

eth2              Link encap:Ethernet  HWaddr 08:00:27:86:8E:F1
                  BROADCAST MULTICAST  MTU:1500  Metric:1
                  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:0 (0.0 b)  TX bytes:2520 (2.4 KiB)

eth3              Link encap:Ethernet  HWaddr 08:00:27:72:67:AA
                  BROADCAST MULTICAST  MTU:1500  Metric:1
                  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
--More--
```

Vemos que el sistema detecta las 4 tarjetas de red, pero que están inactivas a falta de configuración.

Las configuraremos en los siguientes apartados

- eth0

Intentamos levantar la interfaz eth0, que será la que conectará a Internet

Levantamos la interfaz con el comando `ifup`. Este comando se encargará de ejecutar el fichero de configuración de la interfaz, e invocar al comando `route` para realizar el enrutado.

Ejecutamos el comando con la interfaz eth0:

```
[root@router ~]# ifup eth0
Determining IP information for eth0... done.
[root@router ~]# _
```

El comando nos indica que la interfaz se ha activado correctamente. Lo comprobamos con `ifconfig`

```
[root@router ~]# ifconfig | more
eth0      Link encap:Ethernet  HWaddr 08:00:27:C8:CC:DC
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec8:ccdc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2124 (2.0 KiB)  TX bytes:2346 (2.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:264 (264.0 b)  TX bytes:264 (264.0 b)
```

Vemos que la interfaz se ha activado correctamente, con la IP pública 10.0.2.15, así que lanzamos un ping a una web conocida para comprobar la conectividad

```
[root@router ~]# ping google.es
PING google.es (216.58.210.163) 56(84) bytes of data.
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=1 ttl=63 time=20.2 ms
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=2 ttl=63 time=25.6 ms
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=3 ttl=63 time=38.4 ms
^C
--- google.es ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2642ms
rtt min/avg/max/mdev = 20.275/28.128/38.425/7.610 ms
[root@router ~]# _
```

Comprobamos que la conectividad se ha establecido de forma correcta.

- **eth1**

Comprobada la interfaz eth0, repetimos el proceso con la eth1, que será la interfaz que conectara con el equipo anfitrión.

Para ello, desactivamos la interfaz eth0, y activamos la interfaz eth1, para forzar que la comunicación con el anfitrión se realice con dicha interfaz.

Desactivamos la interfaz eth0 con el comando `ifdown`, y activamos la eth1 con `ifup`

```
[root@router ~]# ifdown eth0
[root@router ~]# ifup eth1

Determining IP information for eth1... done.
```

El comando no suelta ningún error, así que comprobamos que la interfaz se ha activado correctamente mediante `ifconfig`

```
[root@router ~]# ifconfig
eth1      Link encap:Ethernet  HWaddr 08:00:27:2D:24:D4
          inet addr:192.168.1.114  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe2d:24d4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:337 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23961 (23.3 KiB)  TX bytes:2622 (2.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:264 (264.0 b)  TX bytes:264 (264.0 b)
```

Vemos que la interfaz se ha activado correctamente, con la IP privada 192.168.1.114

Ahora probaremos la conectividad haciendo ping al equipo anfitrión, para lo cual tendremos que obtener su dirección IP. En este caso, al no disponer del comando `ifconfig`, realizaremos este proceso mediante `ip tools`, con el comando `ip addr`

```
[almu@debian ~]$ ip addr
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: enp0s25: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether ff:ff:ff:ff:ff:ff
    inet 192.168.1.117/24 brd 192.168.1.255 scope global dynamic noprefixroute wlp3s0
        valid_lft 2963sec preferred_lft 2963sec
    inet6 fe80::81c2:b712:b961:6d5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Tras ejecutar este comando, obtenemos que la dirección IP del equipo anfitrión es 192.168.1.117, así que lanzamos un ping desde la máquina virtual a esta dirección

```
[root@router ~]# ping 192.168.1.117
PING 192.168.1.117 (192.168.1.117) 56(84) bytes of data.
64 bytes from 192.168.1.117: icmp_seq=1 ttl=64 time=0.343 ms
64 bytes from 192.168.1.117: icmp_seq=2 ttl=64 time=0.179 ms
64 bytes from 192.168.1.117: icmp_seq=3 ttl=64 time=0.155 ms
64 bytes from 192.168.1.117: icmp_seq=4 ttl=64 time=0.143 ms
64 bytes from 192.168.1.117: icmp_seq=5 ttl=64 time=0.150 ms
^C
--- 192.168.1.117 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4380ms
rtt min/avg/max/mdev = 0.143/0.194/0.343/0.075 ms
```

Comprobamos que tenemos conectividad con el equipo anfitrión.

Lanzamos otro ping en sentido inverso, desde el equipo anfitrión a la máquina virtual

```
[almu@debian ~]$ ping 192.168.1.114
PING 192.168.1.114 (192.168.1.114) 56(84) bytes of data.
64 bytes from 192.168.1.114: icmp_seq=1 ttl=64 time=0.227 ms
64 bytes from 192.168.1.114: icmp_seq=2 ttl=64 time=0.209 ms
64 bytes from 192.168.1.114: icmp_seq=3 ttl=64 time=0.204 ms
64 bytes from 192.168.1.114: icmp_seq=4 ttl=64 time=0.272 ms
^C
--- 192.168.1.114 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.204/0.228/0.272/0.026 ms
```

Comprobamos que tenemos conectividad del equipo anfitrión a la máquina virtual, en ambos sentidos, de forma correcta.

- **eth2 y eth3**

Ya tenemos establecida la conexión a Internet y hacia el equipo anfitrión, en las interfaces eth0 y eth1.

Ahora intentaremos activar las interfaces eth2 y eth3, que serán las que conecten con los equipos clientes Choco y Gamba.

```
[root@router ~]# ifup eth2  
Determining IP information for eth2... failed.
```

En este caso, el comando nos avisa de que no ha logrado encontrar ninguna configuración para la interfaz, y que no ha sido capaz de asignar dirección IP a la interfaz

Repetimos el proceso con eth3, con el mismo resultado

```
[root@router ~]# ifup eth3  
Determining IP information for eth3... failed.
```

Ejecutamos `ifconfig` para ver el estado de las interfaces.

```
eth2      Link encap:Ethernet  HWaddr 08:00:27:86:8E:F1  
          inet6 addr: fe80::a00:27ff:fe86:8ef1/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 b)  TX bytes:5172 (5.0 KiB)  
  
eth3      Link encap:Ethernet  HWaddr 08:00:27:72:67:AA  
          inet6 addr: fe80::a00:27ff:fe72:67aa/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 b)  TX bytes:5256 (5.1 KiB)  
--More--
```

Vemos que, tal como nos avisó el anterior comando, las interfaces están activas pero sin configurar, y que por tanto no tienen dirección IP

En los siguientes apartados realizaremos dicha configuración para poder conectar las interfaces.

## 4. Activando las interfaces eth0 y eth1 al inicio del sistema

En el paso anterior, hemos comprobado que las interfaces eth0 y eth1 tienen buena conectividad, pero que hay que activarlas manualmente, dado que el sistema no las activa de forma automática.

Para cambiar esto, debemos editar su configuración. Dicha configuración se guarda en los ficheros ubicados en */etc/sysconfig/network-scripts/ifcfg-ethX*, siendo X el número de la interfaz.

Abrimos el fichero correspondiente a la interfaz eth0.

```
DEVICE=eth0
HWADDR=08:00:27:C8:CC:DC
TYPE=Ethernet
UUID=6c370a2f-5067-4e8e-9355-113e3581d910
ONBOOT=no
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

Para que la interfaz se inicie durante el arranque, debemos cambiar el valor de ONBOOT a “yes”, así que lo cambiamos

```
DEVICE=eth0
HWADDR=08:00:27:C8:CC:DC
TYPE=Ethernet
UUID=6c370a2f-5067-4e8e-9355-113e3581d910
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

Repetimos el proceso con la interfaz eth1

```
DEVICE=eth1
HWADDR=08:00:27:2D:24:D4
TYPE=Ethernet
UUID=46d0e39d-c5c5-4915-9b45-0d54bdc431ee
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

De momento, el resto de datos los dejamos tal como están.

Tras reiniciar el sistema, vemos que las dos interfaces se han activado correctamente.

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:C8:CC:DC
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec8:ccdc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:590 (590.0 b)  TX bytes:930 (930.0 b)

eth1      Link encap:Ethernet  HWaddr 08:00:27:2D:24:D4
          inet addr:192.168.1.114  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe2d:24d4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1571 (1.5 KiB)  TX bytes:1056 (1.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
--More--
```

## 5. Configurando las interfaces eth2 y eth3

Una vez configurada la conexión a Internet y al anfitrión mediante eth0 y eth1, pasamos a configurar las interfaces eth1 y eth2.

Estas interfaces las configuraremos con IP estática, en dos redes diferentes, que conectarán a las máquinas Choco y Gamba

- **eth2**

Empezamos configurando la interfaz eth2. Esta interfaz estará conectada a la red 192.168.2.0, a la cual también estará conectada la máquina Choco.

Le asignaremos los siguientes parámetros:

- Dirección IP: 192.168.2.1
- Mascara: 255.255.255.0
- Red: 192.168.2.0
- Broadcast: 192.168.2.255

Esta configuración la escribiremos en el fichero de configuración ***/etc/sysconfig/network-scripts/ifcfg-eth2***

Para asignar un direccionamiento estático, cambiaremos el valor de BOOTPROTO a static, y pondremos a continuación el resto de parámetros de nuestra red.

El fichero quedará así:

```
DEVICE=eth2
HWADDR=08:00:27:86:8E:F1
TYPE=Ethernet
UUID=e718fb0e-01a6-41d7-ae4a-7d63e0fec3e2
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.168.2.1
NETMASK=255.255.255.0
NETWORK=192.168.2.0
BROADCAST=192.168.2.255

"/etc/sysconfig/network-scripts/ifcfg-eth2" 13L, 226C
```



Levantamos la interfaz con `ifup`

```
[root@router ~]# ifup eth2
Determining if ip address 192.168.2.1 is already in use for device eth2...
```

Vemos que detecta la configuración de forma correcta.

Ejecutamos `ifconfig eth2` para comprobar el estado de la interfaz

```
[root@router ~]# ifconfig eth2
eth2      Link encap:Ethernet  HWaddr 08:00:27:86:8E:F1
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe86:8ef1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1710 (1.6 KiB)  TX bytes:1446 (1.4 KiB)
```

Vemos que la configuración se ha aplicado correctamente, y la interfaz tiene los parámetros que le hemos asignado en su fichero de configuración.

- **eth3**

Repetimos el proceso con `eth3`. Esta será la interfaz que usemos para conectar con la máquina Gamba, a través de la red 192.168.3.0.

Le asignaremos los siguientes parámetros:

- Dirección IP: 192.168.3.1
- Mascara: 255.255.255.0
- Red: 192.168.2.0
- Broadcast: 192.168.3.255

El fichero quedará así:

```
DEVICE=eth3
HWADDR=08:00:27:72:67:AA
TYPE=Ethernet
UUID=c5ef145e-5ed4-4bd3-b293-e4e5e892c51d
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.168.3.1
NETMASK=255.255.255.0
NETWORK=192.168.3.0
BROADCAST=192.168.3.255
```

Levantamos la interfaz con ifup

```
[root@router ~]# ifup eth3
Determining if ip address 192.168.3.1 is already in use for device eth3...
```

Comprobamos el resultado con ifconfig

```
[root@router ~]# ifconfig eth3
eth3      Link encap:Ethernet  HWaddr 08:00:27:72:67:AA
          inet addr:192.168.3.1  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe72:67aa/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:828 (828.0 b)
```

Vemos que la configuración se ha aplicado correctamente.

## **6. Conectando Choco al router**

Con las interfaces del router ya configuradas, pasamos a configurar la interfaz de red de la máquina Choco, que se conectará a la red 192.168.2.0 usando la dirección IP 192.168.2.2.

La interfaz que configuraremos será la eth0

Le asignaremos los siguientes parámetros:

- Dirección IP: 192.168.2.2
- Mascara: 255.255.255.0
- Red: 192.168.2.0
- Broadcast: 192.168.2.255
- Puerta de enlace: 192.168.2.1

El fichero quedará así:

```
DEVICE=eth0
HWADDR=08:00:27:42:10:E7
TYPE=Ethernet
UUID=188b8c2c-646b-45af-8cf5-14d97bb214b4
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.168.2.2
NETMASK=255.255.255.0
NETWORK=192.168.2.0
BROADCAST=192.168.2.255
GATEWAY=192.168.2.1
```

Levantamos la interfaz usando ifup

```
[root@choco ~]# ifup eth0
Determining if ip address 192.168.2.2 is already in use for device eth0...
```

Comprobamos los cambios con ifconfig

```
[root@choco ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:42:10:E7
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe42:10e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:512 (512.0 b)  TX bytes:1958 (1.9 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:260 (260.0 b)  TX bytes:260 (260.0 b)
```

Vemos que la configuración se ha aplicado correctamente.

Para comprobar la conectividad, lanzamos un ping al router, a través de su interfaz eth2, con dirección 192.168.2.1

```
[root@choco ~]# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.524 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.274 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.276 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.250 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=0.273 ms
64 bytes from 192.168.2.1: icmp_seq=6 ttl=64 time=0.248 ms
^C
--- 192.168.2.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5278ms
rtt min/avg/max/mdev = 0.248/0.307/0.524/0.099 ms
```

Vemos que el ping llega correctamente, lo cual nos indica que la conexión se ha establecido.

Lanzamos otro ping, esta vez desde el router hacia Choco

```
[root@router ~]# ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=64 time=0.327 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=64 time=0.253 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=64 time=0.266 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=64 time=0.266 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=64 time=0.292 ms
^C
--- 192.168.2.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4697ms
rtt min/avg/max/mdev = 0.253/0.280/0.327/0.033 ms
```

Vemos que el ping llega correctamente

## 7. Conectando Gamba al router

Por último, configuramos la interfaz de Gamba para que se conecte al router, a través de la red 192.168.3.0

La interfaz que configuraremos será la eth0

Le asignaremos los siguientes parámetros:

- Dirección IP: 192.168.3.2
- Mascara: 255.255.255.0
- Red: 192.168.3.0
- Broadcast: 192.168.3.255
- Puerta de enlace: 192.168.3.1

El fichero quedará así:

```
DEVICE=eth0
HWADDR=08:00:27:AB:7B:95
TYPE=Ethernet
UUID=99765cd9-5da2-44e3-819a-2ea3899f57f8
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.168.3.2
NETMASK=255.255.255.0
BROADCAST=192.168.3.255
GATEWAY=192.168.3.1
```

Levantamos la interfaz con ifup

```
[root@gamba ~]# ifup eth0
Determining if ip address 192.168.3.2 is already in use for device eth0...
```

Comprobamos la interfaz con ifconfig

```
[root@gamba ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:AB:7B:95
          inet addr:192.168.3.2  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feab:7b95/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:89 errors:0 dropped:0 overruns:0 frame:0
          TX packets:120 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:58689 (57.3 KiB)  TX bytes:9030 (8.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Vemos que la configuración se ha aplicado correctamente, así que probamos la conectividad lanzando un ping al router, a través de la dirección 192.168.3.1

```
[root@gamba ~]# ping 192.168.3.1
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=0.785 ms
64 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=0.269 ms
^C
--- 192.168.3.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1503ms
rtt min/avg/max/mdev = 0.269/0.527/0.785/0.258 ms
```

Vemos que el ping llega correctamente, lo cual nos indica que la configuración es correcta.

Lanzamos otro ping desde el router hacia Gamba

```
[root@router ~]# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=64 time=0.257 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=64 time=0.251 ms
64 bytes from 192.168.3.2: icmp_seq=3 ttl=64 time=0.282 ms
64 bytes from 192.168.3.2: icmp_seq=4 ttl=64 time=0.222 ms
64 bytes from 192.168.3.2: icmp_seq=5 ttl=64 time=0.385 ms
^C
--- 192.168.3.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4596ms
rtt min/avg/max/mdev = 0.222/0.279/0.385/0.058 ms
```

Vemos que el ping llega correctamente.

## 8. Configurando enrutamiento

Tras estos pasos, ya hemos conseguido conectar las máquinas Choco y Gamba con el router. Pero todavía no es posible interconectar ambas máquinas entre sí.

Lo comprobamos lanzando un ping desde Choco (192.168.2.2) hacia Gamba (192.168.3.2)

```
[root@choco ~]# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
^C
--- 192.168.3.2 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5153ms
```

Comprobamos que Choco no es capaz de establecer conexión con Gamba.

```
[root@gamba ~]# ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
^C
--- 192.168.2.2 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4768ms
```

Lanzando un ping de Gamba hacia Choco sucede lo mismo.

Igualmente, ninguna de las dos es capaz de conectar a Internet.

```
[root@choco ~]# ping google.es
ping: unknown host google.es
```

```
[root@gamba ~]# ping google.es
ping: unknown host google.es
```

Para resolver esta situación, tenemos que configurar dentro del router el enrutamiento de paquetes, que nos permitirá redirigir el tráfico de una red hacia otra, redirigiéndolo a través de las interfaces del router.

- **Activando reenvío de paquetes**

El primer paso para configurar el enrutamiento es activar el reenvío de paquetes. Para ello, debemos editar el fichero `/etc/sysctl.conf`.

Abrimos el fichero, y añadimos la línea

```
net.ipv4.conf.default.forwarding=1
```

El fichero quedará así:

```
# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1

# Controls the default maximum size of a message queue
kernel.msgmnb = 65536

# Controls the maximum size of a message, in bytes
kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes
kernel.shmmax = 4294967295

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 268435456

net.ipv4.conf.default.forwarding=1

"/etc/sysctl.conf" 40L, 1092C written
```

Una vez guardada la configuración, reiniciamos el servicio de red para aplicar los cambios

Tras esto, intentamos lanzar un ping desde Choco hasta Gamba

```
[root@choco ~]# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
From 192.168.2.1 icmp_seq=1 Destination Host Prohibited
From 192.168.2.1 icmp_seq=2 Destination Host Prohibited
From 192.168.2.1 icmp_seq=3 Destination Host Prohibited
From 192.168.2.1 icmp_seq=4 Destination Host Prohibited
From 192.168.2.1 icmp_seq=5 Destination Host Prohibited
From 192.168.2.1 icmp_seq=6 Destination Host Prohibited
^C
--- 192.168.3.2 ping statistics ---
6 packets transmitted, 0 received, +6 errors, 100% packet loss, time 5743ms
```

Vemos que el ping llega, pero que el host nos marca el destino como prohibido,

Lo mismo sucede en sentido contrario

```
[root@gamba ~]# ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
From 192.168.3.1 icmp_seq=1 Destination Host Prohibited
From 192.168.3.1 icmp_seq=2 Destination Host Prohibited
From 192.168.3.1 icmp_seq=3 Destination Host Prohibited
From 192.168.3.1 icmp_seq=4 Destination Host Prohibited
From 192.168.3.1 icmp_seq=5 Destination Host Prohibited
From 192.168.3.1 icmp_seq=6 Destination Host Prohibited
^C
--- 192.168.2.2 ping statistics ---
6 packets transmitted, 0 received, +6 errors, 100% packet loss, time 5333ms
```

- **Configurando el NAT en IPTables**

Para resolver esto, debemos configurar las reglas de NAT dentro de IPTables, que permitirá a otros equipos conectarse a través de nuestro router desde sus redes privadas.

```
[root@router ~]# iptables -F
[root@router ~]# iptables -X
[root@router ~]# iptables -Z
[root@router ~]# iptables -t nat -F
[root@router ~]# iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -d 0/0 -j MASQUERADE
[root@router ~]# INPUT ACCEPT
-bash: INPUT: command not found
[root@router ~]# iptables -P INPUT ACCEPT
[root@router ~]# iptables -P FORWARD ACCEPT
[root@router ~]# iptables -t nat -P POSTROUTING ACCEPT
[root@router ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]

[root@router ~]# iptables -t nat -A POSTROUTING -s 192.168.3.0/24 -d 0/0 -j MASQUERADE
[root@router ~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
```

Tras aplicar esta configuración, probamos de nuevo a lanzar el ping desde Choco a Gamba, y en sentido contrario

```
[root@choco ~]# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=63 time=0.521 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=63 time=0.559 ms
64 bytes from 192.168.3.2: icmp_seq=3 ttl=63 time=0.490 ms
64 bytes from 192.168.3.2: icmp_seq=4 ttl=63 time=0.476 ms
^C
--- 192.168.3.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3544ms
rtt min/avg/max/mdev = 0.476/0.511/0.559/0.039 ms
[root@choco ~]# _
```

```
[root@gamba ~]# ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=63 time=0.526 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=63 time=0.537 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=63 time=0.510 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=63 time=0.500 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=63 time=0.570 ms
64 bytes from 192.168.2.2: icmp_seq=6 ttl=63 time=0.437 ms
^C
--- 192.168.2.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 0.437/0.513/0.570/0.044 ms
```

Vemos que, esta vez, ambos ping llegan correctamente y sin restricción



## 9. Asignando IP estática a eth1 en el router

Tras encender la máquina que hace de router, comprobamos que la interfaz eth1 (la interfaz que conecta al anfitrión) no se activa correctamente, y no logra obtener dirección IP mediante dhcp

Para solucionarlo, vamos a asignarle una dirección IP de forma estática. Como esta red debe conectar con el anfitrión, la dirección IP debe ser de la misma red que este.

Hacemos un `ip addr` en el equipo anfitrión, para comprobar su dirección IP.

```
[almu@debian ~]$ ip addr
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: enp0s25: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether [REDACTED] brd ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether [REDACTED] brd ff:ff:ff:ff:ff:ff
    inet 172.17.114.202/19 brd 172.17.127.255 scope global dynamic noprefixroute wlp3s0
        valid_lft 38629sec preferred_lft 38629sec
    inet6 fe80::d4:af5d:8ddc:6450/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Vemos que, esta vez, la dirección IP es 172.17.114.202, con mascara 19, así que debemos asignar una IP del mismo rango a nuestra interfaz.

Le asignaremos los siguientes parámetros:

- Dirección IP: 172.17.114.203
- Mascara: 255.255.224.0

El fichero quedará así:

```
DEVICE=eth1
HWADDR=08:00:27:2D:24:D4
TYPE=Ethernet
UUID=46d0e39d-c5c5-4915-9b45-0d54bdc431ee
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=172.17.114.203
NETMASK=255.255.224.0
BROADCAST=172.17.114.255
```

Levantamos la interfaz con ifup

```
[root@router ~]# ifup eth1
Determining if ip address 192.168.1.2 is already in use for device eth1...
```

Comprobamos la interfaz con ifconfig

```
[root@router ~]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:2D:24:D4
          inet addr:172.17.114.203  Bcast:172.17.114.255  Mask:255.255.224.0
          inet6 addr: fe80::a00:27ff:fe2d:24d4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:34904 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3688257 (3.5 MiB)  TX bytes:7404 (7.2 KiB)
```

Lanzamos un ping al equipo anfitrión para comprobar la conectividad.

```
[root@router ~]# ping 172.17.114.202
PING 172.17.114.202 (172.17.114.202) 56(84) bytes of data.
64 bytes from 172.17.114.202: icmp_seq=1 ttl=64 time=0.221 ms
64 bytes from 172.17.114.202: icmp_seq=2 ttl=64 time=0.123 ms
64 bytes from 172.17.114.202: icmp_seq=3 ttl=64 time=0.143 ms
64 bytes from 172.17.114.202: icmp_seq=4 ttl=64 time=0.140 ms
^C
--- 172.17.114.202 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3804ms
rtt min/avg/max/mdev = 0.123/0.156/0.221/0.040 ms
```

Vemos que tenemos conectividad con el equipo anfitrión, así que la configuración es correcta

## 10. Comprobando conectividad en Choco y Gamba

En los pasos anteriores, hemos comprobado la conectividad entre Choco y Gamba. Ahora realizaremos la misma prueba, pero desde Choco y Gamba hacia el sistema anfitrión y hacia Internet.

Para ello, haremos uso de ping.

Comenzamos lanzando un ping desde Choco al anfitrión.

```
[root@choco ~]# ping 172.17.114.202
PING 172.17.114.202 (172.17.114.202) 56(84) bytes of data.
64 bytes from 172.17.114.202: icmp_seq=1 ttl=63 time=0.894 ms
64 bytes from 172.17.114.202: icmp_seq=2 ttl=63 time=0.391 ms
64 bytes from 172.17.114.202: icmp_seq=3 ttl=63 time=0.430 ms
64 bytes from 172.17.114.202: icmp_seq=4 ttl=63 time=0.333 ms
^C
--- 172.17.114.202 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3311ms
rtt min/avg/max/mdev = 0.333/0.512/0.894/0.223 ms
```

Comprobamos que el ping llega correctamente.

Repetimos el proceso con Gamba, también exitoso

```
[root@gamba ~]# ping 172.17.114.202
PING 172.17.114.202 (172.17.114.202) 56(84) bytes of data.
64 bytes from 172.17.114.202: icmp_seq=1 ttl=63 time=0.426 ms
64 bytes from 172.17.114.202: icmp_seq=2 ttl=63 time=0.474 ms
^C
--- 172.17.114.202 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1865ms
rtt min/avg/max/mdev = 0.426/0.450/0.474/0.024 ms
```

Para probar la conexión a Internet, haremos un ping a una URL conocida.

```
[root@gamba ~]# ping google.es
ping: unknown host google.es
```

Vemos que Gamba no reconoce la URL.

```
[root@choco ~]# ping google.es
ping: unknown host google.es
```

Con Choco pasa lo mismo.

Sospechamos que puede estar desconfigurado el DNS, así que lo comprobamos lanzando un ping a una IP pública

```
[root@choco ~]# ping 216.58.211.227
PING 216.58.211.227 (216.58.211.227) 56(84) bytes of data.
64 bytes from 216.58.211.227: icmp_seq=1 ttl=61 time=16.5 ms
64 bytes from 216.58.211.227: icmp_seq=2 ttl=61 time=15.9 ms
64 bytes from 216.58.211.227: icmp_seq=3 ttl=61 time=16.0 ms
64 bytes from 216.58.211.227: icmp_seq=4 ttl=61 time=16.0 ms
^C
--- 216.58.211.227 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3444ms
rtt min/avg/max/mdev = 15.909/16.130/16.517/0.247 ms
```

```
[root@gamba ~]# ping 216.58.211.227
PING 216.58.211.227 (216.58.211.227) 56(84) bytes of data.
64 bytes from 216.58.211.227: icmp_seq=1 ttl=61 time=16.5 ms
64 bytes from 216.58.211.227: icmp_seq=2 ttl=61 time=15.8 ms
64 bytes from 216.58.211.227: icmp_seq=3 ttl=61 time=16.4 ms
64 bytes from 216.58.211.227: icmp_seq=4 ttl=61 time=15.9 ms
^C
--- 216.58.211.227 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3617ms
rtt min/avg/max/mdev = 15.863/16.199/16.514/0.306 ms
```

Vemos que, efectivamente, el ping llega, así que el problema debe estar en el DNS, y debemos configurarlo.

## 11. Configurando el DNS en Choco y Gamba

Para configurar el DNS, editamos el fichero ifcfg-eth0, añadiendo el parámetro DNS1, seguidos del servidor DNS que queremos añadir. Si quisiéramos añadir mas servidores, los añadiríamos con las líneas DNS2, DNS3... etc, de la misma forma que el ejemplo anterior.

En nuestro caso, solo añadiremos un servidor DNS, con dirección 8.8.8.8

El fichero (en Gamba) quedará así:

```
DEVICE=eth0
HWADDR=08:00:27:AB:7B:95
TYPE=Ethernet
UUID=99765cd9-5da2-44e3-819a-2ea3899f57f8
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.168.3.2
NETMASK=255.255.255.0
BROADCAST=192.168.3.255
GATEWAY=192.168.3.1
DNS1=8.8.8.8
```

Reiniciamos la interfaz:

```
[root@gamba ~]# ifdown eth0
[root@gamba ~]# ifup eth0
Determining if ip address 192.168.3.2 is already in use for device eth0...
```

Probamos a lanzar un ping a una URL conocida.

```
[root@gamba ~]# ping google.es
```

—

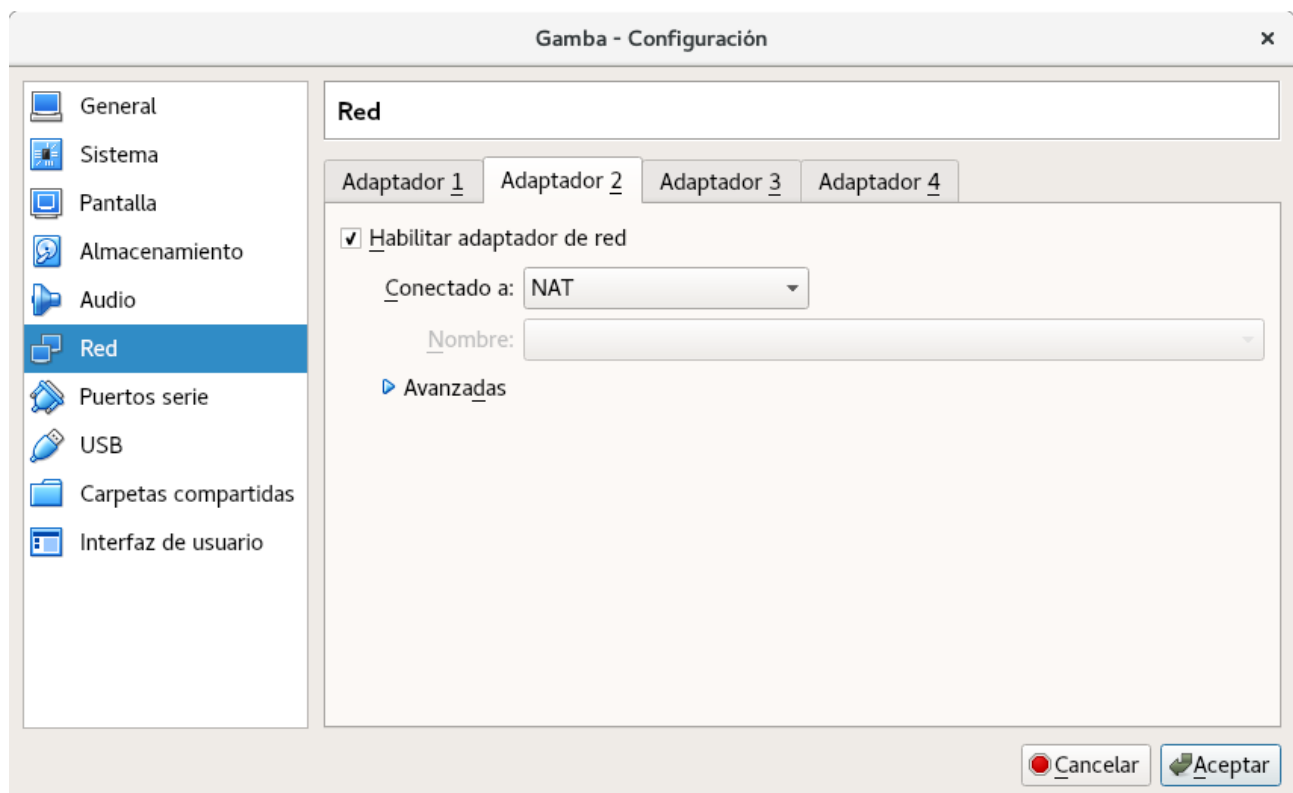
Vemos que el ping sigue sin respuesta, así que seguimos revisando.

Probamos el comando nslookup, para obtener la dirección IP asociada a dicha URL.

```
[root@gamba ~]# nslookup google.es
-bash: nslookup: command not found
```

Vemos que CentOS no encuentra el comando, así que tendremos que instalar el servidor DNS. Dado que no tenemos conexión a Internet para instalar el paquete, tendremos que activar una segunda tarjeta de red, que la conectaremos a internet.

Activamos la tarjeta de red, e iniciamos la máquina



Creamos un fichero de configuración para dicha interfaz, que se conectará mediante dhcp, con estos datos:

```
DEVICE=eth1
BOOTPROTO=dhcp
```

Activamos la interfaz con ifup

```
[root@gamba ~]# ifup eth1
Determining IP information for eth1... done.
```

Una vez levantada la interfaz, ya tenemos acceso a internet, y podemos instalar el paquete correspondiente al servidor DNS, que en este caso se llama bind

```
--> Processing Dependency: libbind9.so.80 for package: 32:bind-9.8.2-0.62.rc1.el6_9.4.i686
--> Running transaction check
---> Package bind-libs.i686 32:9.8.2-0.62.rc1.el6_9.4 will be installed
---> Package portreserve.i686 0:0.0.4-11.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch      Version                               Repository      Size
=====
Installing:
bind                   i686      32:9.8.2-0.62.rc1.el6_9.4           updates         4.0 M
Installing for dependencies:
bind-libs              i686      32:9.8.2-0.62.rc1.el6_9.4           updates         902 k
portreserve            i686      0.0.4-11.el6                        base             23 k
=====

Transaction Summary
=====
Install      3 Package(s)

Total download size: 4.9 M
Installed size: 9.6 M
Is this ok [y/N]: y_
```

Tras este ultimo paso, apagamos la interfaz eth1, y probamos a lanzar un ping a una URL conocida.

```
[root@gamba ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:AB:7B:95
          inet addr:192.168.3.2  Bcast:192.168.29.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feab:7b95/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:50 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2327 (2.2 KiB)  TX bytes:3752 (3.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

```

[root@gamba ~]# ping google.es
PING google.es (216.58.210.163) 56(84) bytes of data.
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=1 ttl=53 time=19.5 ms
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=2 ttl=53 time=37.5 ms
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=3 ttl=53 time=41.7 ms
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=4 ttl=53 time=17.1 ms
^C
--- google.es ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3537ms
rtt min/avg/max/mdev = 17.108/28.994/41.739/10.783 ms
[root@gamba ~]# _

```

Vemos que el ping ya funciona correctamente.

Repetimos los mismos pasos en Choco, y comprobamos que también tenemos conexión.

```

DEVICE=eth0
HWADDR=08:00:27:42:10:E7
TYPE=Ethernet
UUID=188b8c2c-646b-45af-8cf5-14d97bb214b4
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.168.2.2
NETMASK=255.255.255.0
NETWORK=192.168.2.0
BROADCAST=192.168.2.255
GATEWAY=192.168.2.1
DNS1=8.8.8.8

```

```

[root@choco ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:42:10:E7
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe42:10e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:300 (300.0 b)  TX bytes:828 (828.0 b)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

```



```
[root@choco ~]# ifdown eth0
[root@choco ~]# ifup eth0
Determining if ip address 192.168.2.2 is already in use for device eth0...
[root@choco ~]# ping google.es
PING google.es (216.58.210.163) 56(84) bytes of data.
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=1 ttl=53 time=16.7 ms
64 bytes from mad06s10-in-f163.1e100.net (216.58.210.163): icmp_seq=2 ttl=53 time=20.8 ms
^C
--- google.es ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1680ms
rtt min/avg/max/mdev = 16.774/18.827/20.880/2.053 ms
[root@choco ~]#
```

Tras comprobar la conexión a Internet con éxito en ambas máquinas, asumimos que la configuración del DNS es correcta.

## **12. Resumen**

En esta práctica hemos aprendido a simular una infraestructura real usando VirtualBox, incluyendo el router como una máquina virtual más con CentOS.

El proceso de enrutamiento entre dos máquinas requiere que las interfaces de conexión entre ambas queden dentro de la misma subred. Si la máquina servidor no dispone de un servicio DHCP, este proceso lo tendremos que realizar manualmente.

Una vez conectadas las interfaces, si queremos que las máquinas conectadas a las interfaces del router puedan comunicarse entre sí, es necesario configurar el reenvío de paquetes, y establecer las reglas necesarias para que se permita esa comunicación. En el caso de GNU/Linux, estas reglas se establecen a través de IPTables.

En las máquinas cliente, además habrán de configurarse las puertas de enlace, indicando como dirección IP la correspondiente a la interfaz del router a la que está conectado (que debe estar en la misma red que el cliente)

Con esto ya tendremos las máquinas cliente interconectadas a través del router, pero todavía serán incapaces de reconocer la IP asociada a una URL. Para ello, tendremos que configurar el servidor de nombres (o DNS) dentro de cada una de las máquinas, indicando dentro de su configuración de red la dirección IP del servidor de nombres que queremos usar.

Y, finalmente, con este último paso, tendremos establecida la conexión entre todas las máquinas.