

# CONFIGURACIÓN TRADICIONAL DE LA RED EN LINUX

 [Descargar PDF](#)

## ÍNDICE

- INTRODUCCIÓN
- MOSTRAR CONFIGURACIÓN INTERFACES DE RED
- ▼ CONFIGURACIÓN ESTÁTICA
  - ▼ CONFIGURAR IP Y GATEWAY
    - CASO NO PERSISTENTE
    - ▼ CASO PERSISTENTE
      - SISTEMAS BASADOS EN DEBIAN
      - SISTEMAS BASADOS EN RED HAT
  - ▼ CONFIGURAR DNS
    - SISTEMAS BASADOS EN DEBIAN
    - SISTEMAS BASADOS EN RED HAT
  - BORRAR IP Y GATEWAY
  - BORRAR DNS
- ▼ CONFIGURACIÓN DINÁMICA
  - OBTENER PARÁMETROS DE RED
  - LIBERAR IP
  - CASO PERSISTENTE
- ACTIVAR Y DESACTIVAR INTERFACES DE RED
- ▼ TABLA DE ENRUTAMIENTO
  - MOSTRAR TABLA
  - ▼ MODIFICAR TABLA
    - AÑADIR ENTRADA
    - BORRAR ENTRADA
    - PERSISTENCIA
- ▼ TABLA ARP
  - MOSTRAR TABLA
  - ▼ MODIFICAR TABLA
    - AÑADIR ENTRADA
    - BORRAR ENTRADA

# INTRODUCCIÓN

En el presente tema vamos a estudiar la configuración tradicional de la red en Linux. Nos centraremos en sistemas sin entorno de escritorio ya que en Linux hay una gran variedad de entornos de escritorio y la configuración de la red depende de cada uno de ellos.

Básicamente distinguiremos dos formas diferentes de configuración de la red:

- **Configuración no persistente** o de bajo nivel. Esta configuración se refiere a los cambios realizados directamente desde la línea de comandos, que afectan a la configuración de la red de manera inmediata pero *no se mantienen tras reiniciar el sistema*. Este tipo de configuración es útil para pruebas, cambios temporales o en entornos donde no se requiere persistencia. Se realiza desde la línea de comandos mediante utilidades específicas incluidas en la librería **net-tools**. Las más habituales son `ifconfig`, `route` y `arp`, que permiten gestionar interfaces, direcciones IP y rutas de forma temporal entre otras cosas.
- **Configuración persistente**. La configuración persistente se lleva a cabo en ficheros. Las directivas u órdenes del fichero se ejecutan cuando se inicia la máquina, el servicio de red o el usuario invoca los comandos `ifup` e `ifdown`.

## MOSTRAR CONFIGURACIÓN INTERFACES DE RED

La utilidad o comando `ifconfig` es muy versátil y se puede utilizar para mostrar información de las interfaces de red.

Para mostrar la configuración de las interfaces de red el comando tiene la siguiente sintaxis:

```
ifconfig [-a] [<interfaz>]
```

Entre la información mostrada podemos ver la IP, la MAC, la máscara de red, la dirección de broadcast e información sobre el estado de la interfaz entre otros datos.

Cuando no se especifica un interfaz muestra información de todas las interfaces de red o solo de las activas si omitimos el flag `-a`. Cuando se especifica una interfaz muestra solamente información de esa interfaz (esté o no activa).



### Mostrar interfaces de red

Mostrar el estado de la interfaz `eth0`.

```
root@Host:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:d0:2f:54
          inet addr:192.168.2.16  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed0:2f54/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:26 errors:0 dropped:0 overruns:0 frame:0
          TX packets:88 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3553 (3.4 KiB)  TX bytes:5485 (5.3 KiB)
```

La ejecución del comando nos indica que la interfaz `eth0` tiene la ip `192.168.2.16`, la máscara de red es la `255.255.255.0` y que está activa ( `UP ... RUNNING` ) entre otras cosas.

La sintaxis del comando para mostrar las estadísticas de todas las interfaces (o una concreta) activas es la siguiente:

```
ifconfig [-s] [<interfaz>]
```

## CONFIGURACIÓN ESTÁTICA

Para llevar a cabo la configuración no persistente de las interfaces de red también utilizaremos el comando `ifconfig` además del comando `route`.

Para el caso de la configuración persistente se usan ficheros. Hay que distinguir entre los sistemas basados en **Debian** y los basados en **RedHat**. Nosotros principalmente nos centraremos en los primeros.

## CONFIGURAR IP Y GATEWAY

### CASO NO PERSISTENTE

Para cambiar la dirección IP de una interfaz de red, o asignarle una si no tenía, la sintaxis sería la siguiente:

```
ifconfig <interfaz> {<ip/prefix> | <ip> netmask <máscara>}
```

#### Configurar dirección IP

Cambiar la dirección IP de la interfaz `eth0` a `192.168.2.20/27`.

```
root@Host:~# ifconfig eth0 192.168.2.20/27
```

También se obtendría el mismo resultado con el siguiente comando:

```
root@Host:~# ifconfig eth0 192.168.2.20 netmask 255.255.255.224
```

Para configurar la puerta de enlace o gateway, de forma no persistente, añadiremos una ruta por defecto mediante el comando `route`.

La sintaxis del comando para este propósito es la siguiente:

```
route add default gw <puerta_de_enlace> [metric <métrica>] [dev] [<interfaz>]
```

En el comando se especifica la ruta por defecto ( `default` ) y la IP de la `puerta de enlace` que me proporciona acceso externo (Internet). La opción `dev` se puede omitir si sólo hay una interfaz de red; es más se puede omitir hasta el nombre de la interfaz.

#### Configurar gateway

Configurar la puerta de enlace con la ip `192.168.2.1`. Supondremos que la única interfaz de red es `eth0`.

```
root@Host:~# route add default gw 192.168.2.1
```

## CASO PERISTENTE

Previamente a pasar a describir la configuración persistente de la red, se debe distinguir entre los conceptos de **interfaz física de red** e **interfaz lógica de red**. Un equipo tendrá tantas interfaces físicas de red como tarjetas de red tenga, ya sean reales o virtuales, pero puede tener una cantidad arbitraria de interfaces lógicas de red. A las interfaces lógicas de red también se les denomina **perfiles de conexión de red** o, simplemente, **conexiones de red**.

A partir de ahora, a las interfaces físicas de red las llamaremos simplemente **interfaces** y a las interfaces lógicas **conexiones**.

Una **conexión activa** es aquella cuyos datos de configuración se han asignado a una interfaz de red. Podemos tener tantas conexiones definidas como queramos, pero solo puede haber una única conexión activa por cada interfaz en un momento dado.

Es habitual que las conexiones se llamen igual que las interfaces de red. Por eso a veces se habla indistintamente de interfaces y conexiones.

Cuando se realiza la configuración persistente de la red realmente estamos configurando **conexiones** y no **interfaces**.

## SISTEMAS BASADOS EN DEBIAN

En los sistemas **Debian** y derivados (Ubuntu, ...), especialmente los que no tienen entorno de escritorio, la configuración de la red se lleva a cabo en el archivo `/etc/network/interfaces`.

El archivo de configuración, en su versión más simple, consiste en varias secciones como las que se muestran a continuación:

```
#/etc/network/interfaces
...
auto <lista-interfaces>
...
allow-hotplug <lista-interfaces>
...
# Configuración conexión estática
iface <conexión> inet static
    address <ip>
    netmask <máscara>
    [gateway <puerta-enlace>]
...
```

En la cláusula `auto` tenemos una lista separada por espacios en blanco de interfaces de red que se activarán cuando se ejecuta un comando de activación de la interfaz (cuando se inicia el sistema se activan todas las interfaces de la lista). Puede haber más de una línea con una cláusula `auto`.

En la cláusula `allow-hotplug` tenemos una Lista separada por espacios en blanco de interfaces de red que se activarán cuando se detecta la conexión física de un adaptador de red al equipo (al inicio pasa esto). Por ejemplo, cuando se conecta el cable de red. Puede haber más de una línea con una cláusula `allow-hotplug`.

Mediante la cláusula `iface` llevamos a cabo la configuración de una conexión de red. Se debe indicar el nombre de la conexión ( `<conexión>` ), es obligatorio proporcionar la dirección IP ( `address <ip>` ) y la máscara de red ( `netmask <máscara>` ). La puerta de enlace ( `gateway <puerta-enlace>` ) es opcional, aunque si no la ponemos no tendremos salida a internet. Puede haber más de una sección `iface`.

Es habitual que las conexiones se llamen igual que las interfaces de red. Así, cuando se habla de configuración de la interfaz, lo que realmente se quiere decir es configuración de una conexión que tiene el mismo nombre que la interfaz. Los siguientes ejemplos aclaran los conceptos.

### Configuración red estática Debian

Configurar la interfaz `enp0s3` (realmente es una conexión) de manera estática con la IP `192.168.5.96/26` con puerta de enlace `192.168.5.65`. La conexión se debe activar automáticamente cuando se ejecuta una instrucción de activación de la interfaz.

```
#/etc/network/interfaces
auto enp0s3

# Configuración conexión estática
iface enp0s3 inet static
    address 192.168.5.96
    netmask 255.255.255.192
    gateway 192.168.5.65
```

### Configuración red estática Debian

Configurar una conexión llamada `trabajo` de manera estática con la IP `172.16.100.10/16` con puerta de enlace `172.16.0.1`.

```
#/etc/network/interfaces

# Configuración conexión estática
iface trabajo inet static
    address 172.16.100.10
    netmask 255.255.0.0
    gateway 172.16.0.1
```

## SISTEMAS BASADOS EN RED HAT

En estos sistemas, las interfaces lógicas de red o conexiones se configuran dentro de la carpeta `/etc/sysconfig/network-scripts`. Para cada interfaz lógica o conexión existe un archivo de configuración. La nomenclatura de los archivos de configuración es la siguiente: `ifcfg-<nombre-conexión>`.

Como ocurría con los sistemas basados en Debian, es muy habitual que las conexiones se llamen igual que las interfaces. Así, si tenemos una tarjeta física de red que se llama `eth0`, tendremos una conexión que se llama `ifcfg-eth0`.

Un archivo de configuración de una interfaz consiste en una lista de directivas de la forma `<parámetro>=<valor>`. El archivo de configuración habitual para las conexiones dinámicas y estáticas sigue el siguiente esquema:

```
DEVICE=<interfaz>
ONBOOT={yes|no}
BOOTPROTO=none
IPADDR=<ip>
NETMASK=<máscara-red>
GATEWAY=<puerta-enlace>
```

El parámetro `DEVICE` especifica la interfaz física de red a la cual se asociará la conexión.

Con la directiva `BOOTPROTO=none` indicamos que la conexión se configura de manera estática o manual.

El parámetro `ONBOOT` sirve para que se active, o no, la conexión cuando se inicie el servicio de red o se reinicie la máquina. Solo debe haber una conexión con valor `yes` para cada interfaz de red (parámetro `DEVICE`).

Las siguientes directivas son exclusivas de conexiones estáticas:

- `IPADDR` : contiene la IP que tomará la interfaz asociada cuando se active la conexión en el caso de configuración estática.
- `NETMASK` : es la máscara de red.
- `GATEWAY` : es la puerta de enlace.

Cuando se hace la instalación del sistema operativo, automáticamente se crea un archivo de conexión correspondiente a la primera tarjeta de red.

### Configuración red estática Red Hat

Configurar una conexión llamada `eth2` de manera que la IP sea `192.168.4.10`, la máscara de red `255.255.255.0` y la puerta de enlace `192.168.4.1`. La interfaz asociada a esta conexión es `eth2` y se debe activar cuando se inicia el equipo.

El archivo se llamará `/etc/sysconfig/network-scripts/ifcfg-eth2` y su contenido será el siguiente:

```
DEVICE=eth2
ONBOOT=yes
BOOTPROTO=none
IPADDR=192.168.4.10
NETMASK=255.255.255.0
GATEWAY=192.168.4.1
```

### Configuración red estática Red Hat

Configurar una conexión llamada `trabajo` de manera estática con la IP `172.16.100.10/16` con puerta de enlace `172.16.0.1`. La conexión estará asociada a la interfaz `eth1` y no se iniciará automáticamente.

El archivo se llamará `/etc/sysconfig/network-scripts/ifcfg-trabajo` y su contenido será el siguiente:

```
DEVICE=eth1
ONBOOT=no
BOOTPROTO=none
IPADDR=172.16.100.10
NETMASK=255.255.0.0
GATEWAY=172.16.0.1
```

## CONFIGURAR DNS

La configuración del DNS sólo puede ser **persistente**.

## SISTEMAS BASADOS EN DEBIAN

En los sistemas basados en Debian, la configuración del DNS se realiza editando el archivo `/etc/resolv.conf`. Este archivo contiene una lista de servidores DNS que el sistema utilizará para resolver nombres de dominio. La sintaxis del archivo es la siguiente:

```
nameserver <ip-servidor-dns>
```

Se pueden especificar múltiples servidores DNS (hasta un máximo de 3), uno por línea. Por ejemplo:

```
nameserver <ip-servidor-dns1>
nameserver <ip-servidor-dns2>
```

### Configuración DNS Debian

En una red tenemos un servidor DNS de IP `192.168.3.3`. Configurar una máquina de la red para que los nombres DNS los resuelva el servidor interno, y si este es incapaz de resolverlos, la petición la resuelva el servidor DNS público de IP `8.8.8.8`.

```
nameserver 192.168.3.3
nameserver 8.8.8.8
```

Cuando tenemos instalada la aplicación `resolvconf`, se pueden configurar los servidores DNS en el mismo archivo de configuración de las interfaces de red con la directiva `dns-nameservers`. Podemos comprobar si el servicio está instalado ejecutando, por ejemplo, el comando `service` o `dpkg`. El siguiente comando nos indica si el servicio `resolvconf` está instalado y su estado (activo o inactivo):

```
root@linux:~# service resolvconf status
```

El siguiente comando es equivalente:

```
root@linux:~# dpkg -l | grep resolvconf
```

Además, en este caso el archivo `/etc/resolv.conf` no es realmente el archivo de configuración de los DNS, sino un enlace simbólico (link) al archivo de configuración.

El servicio `resolvconf` está instalado por defecto en muchas distribuciones Ubuntu antiguas como Ubuntu 16.04.

Cuando se activa una conexión de red donde aparece una directiva `dns-nameservers`, el servicio `resolvconf` lee los valores de los servidores DNS y los pone en el archivo `/etc/resolv.conf`. Por tanto, si tenemos instalado este servicio, no debemos configurar los servidores DNS utilizados en el archivo `/etc/resolv.conf`. No obstante, se pueden añadir servidores DNS concretos si los ponemos en el archivo `/etc/resolvconf/resolv.conf.d/head` con una directiva `nameserver`. En este caso, `resolvconf` siempre pondrá este DNS en el archivo `/etc/resolv.conf`, además de los que aparecen en las directivas `dns-nameservers`.

### Configuración DNS resolvconf

Tenemos un equipo con Debian 11 que utiliza el servicio `resolvconf` como cliente DNS. La dirección de red es `192.168.4.0/24` y la IP del equipo es `192.168.4.5`. Queremos configurar la interfaz de red para que utilice los DNS `10.10.10.3` y `8.8.8.8`.

El servicio `resolvconf` no viene instalado por defecto en las distribuciones Debian sin entorno de escritorio. Lo primero que debemos hacer es instalarlo:

```
root@linux:~# apt install resolvconf
```

Ahora ya podemos configurar los servidores DNS utilizados.

```
auto enp0s3
iface enp0s3 inet static
address 192.168.4.5
netmask 255.255.255.0
gateway 192.168.4.1
dns-nameservers 10.10.10.3 8.8.8.8
```

## SISTEMAS BASADOS EN RED HAT

En estos sistemas la configuración del DNS se lleva a cabo en los archivos de configuración de las conexiones. Cláusulas del tipo `DNS<n> = <ip-dns>`, donde `n` indica el número de servidor DNS (el 1 es el más prioritario) se utilizan para definir los servidores DNS.



### Configuración red con DNS Red Hat

Configurar una conexión llamada `eth2` de manera que la IP sea `192.168.4.10`, la máscara de red `255.255.255.0` y la puerta de enlace `192.168.4.1`. Además, los servidores DNS serán el `10.10.10.3` y `8.8.8.8`. La interfaz se debe activar cuando se inicia el equipo.

El archivo se llamará `/etc/sysconfig/network-scripts/ifcfg-eth2` y su contenido será el siguiente:

```
DEVICE=eth2
ONBOOT=yes
BOOTPROTO=none
IPADDR=192.168.4.10
NETMASK=255.255.255.0
GATEWAY=192.168.4.1
DNS1=10.10.10.3
DNS2=8.8.8.8
```

## BORRAR IP Y GATEWAY

Para borrar una ip de manera **no persistente** se asigna la ip `0.0.0.0` a la interfaz de red utilizando el comando `ifconfig`. Cuando se asigna esta ip a una interfaz, la interfaz queda administrativamente activa pero sin ninguna ip. Además, la ruta por defecto asociada a esa interfaz se elimina automáticamente de la tabla de enrutamiento.

```
ifconfig <interfaz> 0.0.0.0
```



Para eliminar la puerta de enlace, sin eliminar la ip de la interfaz, se utiliza el comando `route del` para borrar la ruta asociada.

```
route del default [gw <puerta_de_enlace>] [<interfaz>]
```

Si sólo tenemos una única ruta por defecto (lo habitual) no es necesario especificar la puerta de enlace (ni la interfaz de red).

### Eliminar gateway

Eliminar la ruta por defecto o gateway de la tabla de enrutamiento de un equipo. Suponemos que sólo hay una única ruta por defecto.

```
root@Host:~# route del default
```

Para eliminar la ip en el caso de la configuración **persistente**, sólo hay que desactivar la interfaz, eliminar la sección de configuración de la interfaz (o el archivo de interfaz) y volver a activar la interfaz.

Si sólo queremos eliminar la puerta de enlace tenemos que desactivar la interfaz, eliminar del archivo de configuración la línea donde aparece la directiva `gateway` y volver a activar la interfaz.

## BORRAR DNS

En el caso de los **sistemas basados en Debian** que configuran el DNS en el archivo `/etc/resolv.conf` simplemente hay que editar el archivo y eliminar las líneas correspondientes a los servidores DNS que queremos borrar.

Si se utiliza el servicio resolvconf en conjunción con el archivo `/etc/network/interfaces` sólo tenemos que parar el servicio, editar dicho archivo y eliminar los DNS que nos interesen de las cláusulas `dns-domainserver` y volver a iniciar el servicio resolvconf.

Para los **sistemas basados en Red Hat** hay que desactivar la conexión que contiene los DNS a borrar, editar dicho archivo de conexión y eliminar las líneas que nos interesen donde aparece la palabra DNS. Por último hay que volver a activar la conexión.

## CONFIGURACIÓN DINÁMICA

### OBTENER PARÁMETROS DE RED

Para configurar dinámicamente una interfaz de red; es decir para hacer que la dirección ip y otros parámetros de red (básicamente la puerta de enlace y servidores DNS) nos los proporcione un servidor DHCP no nos sirve el comando `ifconfig`. En su lugar tenemos que utilizar el cliente DHCP que venga con la distribución de linux utilizada. En linux Debian y derivados este cliente se llama `dhclient`.

La sintaxis del comando para obtener la ip dinámicamente es la siguiente:

```
dhclient [-v] <interfaz>
```

El comando hace que la interfaz de red especificada solicite una dirección IP a un servidor DHCP de la red. Con la opción `-v` nos muestra información del proceso de obtención de la dirección.



## Obtener ip dinámicamente

Solicitar una ip dinámicamente para la interfaz de red `eth0` . Mostrar información adicional relativa a la obtención de la dirección.

```
root@Host:~# dhclient -v eth0
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Listening on LPF/eth0/08:00:27:d0:2f:54
Sending on LPF/eth0/08:00:27:d0:2f:54
Sending on Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 4
DHCPPREQUEST on eth0 to 255.255.255.255 port 67
DHCPOFFER from 192.168.2.3
DHCPACK from 192.168.2.3
bound to 192.168.2.16 -- renewal in 299 seconds.
```

Los detalles de la información mostrada se explicaran en otro tema. La última línea me indica que la ip asignada a la interfaz `eth0` es la `192.168.2.16` .

Si a una interfaz ya se le ha asignado una ip dinámicamente, la ejecución del comando provoca la renovación de la concesión de la ip.



## Obtener ip dinámicamente

Solicitar una ip dinámicamente para la interfaz de red `eth0` . Mostrar información adicional relativa a la obtención de la dirección.

```
root@Host:~# dhclient -v eth0
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Listening on LPF/eth0/08:00:27:d0:2f:54
Sending on LPF/eth0/08:00:27:d0:2f:54
Sending on Socket/fallback
DHCPPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.2.3
RTNETLINK answers: File exists
bound to 192.168.2.16 -- renewal in 238 seconds.
```

Como en el ejemplo anterior, la ip asignada a la interfaz `eth0` es la `192.168.2.16` . La ausencia de las líneas que contienen la palabra `DHCPDISCOVER` y `DHCPOFFER` nos indican que la interfaz ya tenía asignada esta ip y que lo que se ha producido es la renovación de la ip.

## LIBERAR IP

Cuando en el comando `dhclient` utilizamos la opción `-r` lo que hacemos es eliminar la ip de nuestro adaptador de red e indicarle al servidor DHCP que la ip está disponible para asignarla a otro equipo.

La sintaxis del comando para este caso es la siguiente:

```
dhclient -r [-v] <interfaz>
```

Libera la ip asignada a la interfaz de red especificada. La interfaz queda administrativamente activa pero sin ninguna ip. Con la opción `-v` se nos muestra información adicional. Si solo tenemos una única interfaz de red podemos omitir el parámetro `<interfaz>`.

### Liberar ip

Libera la ip asignada a la interfaz de red `eth0`. Mostrar información adicional relativa a la liberación de la dirección. Suponemos que solo hay una única interfaz de red.

```
root@Host:~# dhclient -r -v eth0
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Listening on LPF/eth0/08:00:27:d0:2f:54
Sending on LPF/eth0/08:00:27:d0:2f:54
Sending on Socket/fallback
DHCPRELEASE on eth0 to 255.255.255.255 port 67
```

Como solo hay una única interfaz de red, no es necesario especificar el parámetro `eth0`.

```
root@Host:~# dhclient -r -v
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Listening on LPF/eth0/08:00:27:d0:2f:54
Sending on LPF/eth0/08:00:27:d0:2f:54
Sending on Socket/fallback
DHCPRELEASE on eth0 to 255.255.255.255 port 67
```

## CASO PERSISTENTE

Lo visto hasta ahora es para el caso de una configuración no persistente. Para la configuración persistente, en el caso de las **distribuciones basadas en Debian**, solo tenemos que editar el archivo de configuración de las interfaces ( `/etc/network/interfaces` ) e indicar en la directiva `iface` que la asignación es por `dhcp`. La sintaxis general para esta configuración es la siguiente:

```
# /etc/network/interfaces
...

# Configuración conexión dinámica
iface <conexión> inet dhcp
...
```

### Configuración red dinámica Debian

Configurar la interfaz `enp0s3` (realmente es una conexión) de manera dinámica con DHCP.

```
# /etc/network/interfaces
auto enp0s3

# Configuración conexión dinámica
iface enp0s3 inet dhcp
```

Para el caso de las **distribuciones basadas en Red Hat**, pondremos `dhcp` en el valor del parámetro `BOOTPROTO` en el archivo de configuración de la conexión que queremos configurar de esta manera.

La sintaxis general de un archivo de configuración de conexión en este caso es la siguiente:

```
DEVICE=<interfaz>
ONBOOT={yes|no}
BOOTPROTO=dhcp
```

### Configuración red dinámica Red Hat

Configurar una conexión llamada `eth2` de manera que la IP y otros parámetros de red se asignen dinámicamente a la interfaz `eth2`.

El archivo se llamará `/etc/sysconfig/network-scripts/ifcfg-eth2` y su contenido será el siguiente:

```
DEVICE=eth2
ONBOOT=yes
BOOTPROTO=dhcp
```

## ACTIVAR Y DESACTIVAR INTERFACES DE RED

Para activar o desactivar una interfaz de red, de manera **no persistente**, se utiliza el comando `ifconfig`. La sintaxis del comando, en este caso, es la siguiente:

```
ifconfig <interfaz> [up|down]
```

La opción `up` se utiliza para activar la interfaz y la opción `down` para desactivarlo.

### Activar interfaz de red (no persistente)

Activar la interfaz `eth0`.

```
root@Host:~# ifconfig eth0 up
```

Para la activación y desactivación de las interfaces que se han configurado de **manera persistente** se utilizan los comandos `ifup` e `ifdown` respectivamente, los cuales forman parte de la librería o paquete `ifupdown`.

La sintaxis de estos puede variar en función del tipo de distribución y la versión. Para el caso de las **distribuciones basadas en Debian** la sintaxis del comando `ifup` es la siguiente:

```
ifup <interfaz[=conexión]>
```

La ejecución del comando activa la `<interfaz>` especificada con los datos de la `<conexión>` especificada. Cuando no se especifica la conexión, asume que hay una conexión configurada con el mismo nombre que la interfaz. Recordemos que las conexiones están configuradas en el archivo `/etc/network/interfaces` en las cláusulas `iface`.

### Activar interfaz de red (persistente)

En una máquina Debian tenemos configurada de manera persistente las conexiones `eth0` y `universidad`. Activar las interfaces de red `eth0` con la conexión del mismo nombre y la interfaz `eth1` con la conexión `universidad`.

```
root@debian:~# ifup eth0
root@debian:~# ifup eth1=universidad
```

Alternativamente, la interfaz `eth0` se podría activar con el siguiente comando:

```
root@debian:~# ifup eth0=eth0
```

El comando `ifup` se puede utilizar sin especificar la interfaz. En este caso tenemos dos posibilidades:

```
ifup {-a \ | --allow=auto}
```

Se activan todas las interfaces que tengan una directiva `auto` en el archivo de configuración de las interfaces. Cuando se inicia el servicio de red (o se inicia la máquina), se ejecuta este comando automáticamente.

```
ifup [--allow=hotplug]
```

Se activan todas las interfaces que tengan una directiva `allow-hotplug` en el archivo de configuración de las interfaces.

En ambos casos, para que se active una interfaz tiene que haber una conexión del mismo nombre definida en el archivo de configuración de las interfaces.

Para que se active automáticamente una interfaz y se asocie a una conexión con nombre diferente al de la interfaz se pueden utilizar las cláusulas `mapping`. Esto, por ejemplo, nos permitiría asociar una interfaz `eth0` a una conexión llamada `casa` cuando nos conectamos a la red de casa con nuestro portátil y asociar la conexión llamada `trabajo` a la misma interfaz cuando nos conectemos a la red del trabajo.

Las cláusulas `mapping` están fuera del alcance de este curso. En cualquier caso se puede encontrar más información sobre este aspecto y otros no tratados en el siguiente enlace: [Configuración de la red en Debian](#).

La sintaxis del comando `ifdown` es la siguiente:

```
ifdown <interfaz>
```

La ejecución del comando desactiva la `<interfaz>` de red especificada.



## Desactivar interfaz de red (persistente)

En una máquina Debian tenemos configurada de manera persistente las conexiones *eth0* y *universidad*. Desactivar las interfaces de red `eth0` asociada a la conexión del mismo nombre y la interfaz `eth1` asociada a la conexión *universidad*.

```
root@debian:~# ifdown eth0
root@debian:~# ifdown eth1
```

Los comandos `ifup` e `ifdown` tienen una sintaxis ligeramente diferente en los **sistemas basados en Red Hat**. En este caso siempre especificamos como parámetro la `<conexión>`.

Como en el archivo de conexión el parámetro `DEVICE` nos indica con que interfaz se asocia la conexión, los comandos `ifup` e `ifdown` saben a que interfaz hay que asociar una conexión o que interfaz hay que desactivar.

Para activar una conexión ponemos:

```
ifup <conexion>
```

Y para desactivarla:

```
ifdown <conexion>
```



## Desactivar conexión

En una máquina con el sistema Red Hat, desactivar la conexión de nombre *universidad*.

```
root@redhat:~# ifdown universidad
```

El archivo donde está configurada la conexión se llamará `/etc/sysconfig/network-scripts/ifcfg-universidad`. Dentro de este archivo habrá un parámetro `DEVICE` con la interfaz a la cual se asocia la conexión, por ejemplo `DEVICE=eth1`. El comando, realmente, desactiva esta interfaz.

# TABLA DE ENRUTAMIENTO

La utilidad o comando `route` se utiliza para configurar, de **manera no persistente**, y mostrar información de la tabla de enrutamiento de un equipo; es decir la tabla nos indica como llegar a todas las redes: a las que estoy directamente conectado y a otras redes a las que no se está directamente conectado y a Internet.

## MOSTRAR TABLA

```
route [-n]
```

Muestra la tabla de enrutamiento del sistema. Si se utiliza el flag `-n` muestra las direcciones IP en lugar de los nombres de host. La información más relevante que muestra este comando es:

- *Destination*: La red de destino a la que se quiere llegar.

- **Gateway:** La puerta de enlace a través de la cual se accede a la red de destino.
- **Genmask:** La máscara de red que define el rango de direcciones IP de la red de destino.
- **Flags:** Indicadores que muestran el estado de la ruta (U: ruta activa, G: puerta de enlace, H: host, etc.).
- **Metric:** La métrica de la ruta, que indica la preferencia de la ruta (rutas con menor métrica son preferidas).
- **Iface:** La interfaz de red a través de la cual se accede a la red de destino.

## Mostrar tabla de enrutamiento

Mostrar la tabla de enrutamiento del sistema.

```
root@Host:~# route
Kernel IP routing table
Destination    Gateway      Genmask      Flags Metric Ref    Use Iface
default        192.168.2.1  0.0.0.0      UG    0      0      0 eth0
192.168.2.0    *           255.255.255.0 U    0      0      0 eth0
192.168.3.0    192.168.2.2 255.255.255.0 UG    0      0      0 eth1
```

El resultado muestra que la ruta por defecto ( `default` ) utiliza la puerta de enlace `192.168.2.1` a través de la interfaz `eth0` para acceder a Internet.

Si utilizamos el flag `-n` el destino de la ruta por defecto se muestra como una dirección IP ( `0.0.0.0` ) en lugar de `default` . El asterisco en la columna *Gateway* correspondiente a la segunda entrada nos indica que el equipo está directamente conectado a la red `192.168.2.0/24` (a través de la interfaz `eth0` ).

La última entrada nos indica que para acceder a la red `192.168.3.0/24` , el equipo debe utilizar la puerta de enlace `192.168.2.2` a través de la interfaz `eth1` .

```
root@Host:~# route -n
Kernel IP routing table
Destination    Gateway      Genmask      Flags Metric Ref    Use Iface
0.0.0.0        192.168.2.1  0.0.0.0      UG    0      0      0 eth0
192.168.2.0    0.0.0.0      255.255.255.0 U    0      0      0 eth0
192.168.3.0    192.168.2.2 255.255.255.0 UG    0      0      0 eth1
```

## MODIFICAR TABLA

También utilizaremos el comando `route` para añadir entradas o rutas en la tabla de enrutamiento y para eliminarlas. Para cambiar una entrada de la tabla de enrutamiento lo que haremos será eliminar la entrada que queremos cambiar y añadir una nueva con los cambios que queremos hacer.

## AÑADIR ENTRADA

Podemos añadir entradas a la tabla de enrutamiento para acceder a redes a las que no se está directamente conectado. Un caso especial, es la ruta por defecto o ruta que nos da acceso a Internet. Este último caso ya lo vimos anteriormente cuando estudiamos como configurar el *gateway*.

Para añadir una ruta a una red a la que no estamos directamente conectados el comando adopta la siguiente sintaxis:

```
route add -net {<red> netmask <máscara> | <red/prefijo>} gw <puerta_de_enlace> [metric <métrica>] dev [<interfaz>]
```

Añadir una ruta estática a la tabla de enrutamiento. La opción `dev` se puede omitir si sólo hay una interfaz de red; es más se puede omitir hasta el nombre de la interfaz. La opción `metric` permite establecer la métrica de la

ruta, que influye en la selección de rutas.

### Añadir ruta estática

Añadir una ruta estática para acceder a la red `192.168.11.0/25` desde un equipo de la red `192.168.10.0/24` utilizando como puerta de enlace el equipo con IP `192.168.10.2`. El equipo sólo tiene una interfaz de salida de nombre `eth0`.

```
root@Host:~# route add -net 192.168.11.0 netmask 255.255.255.128 gw 192.168.10.2 dev eth0
```

Otra forma alternativa y simplificada del mismo comando sería:

```
root@Host:~# route add -net 192.168.11.0/25 gw 192.168.10.2
```

## BORRAR ENTRADA

De forma análoga a como se añadían rutas, podemos eliminarlas. La sintaxis del comando para eliminar una ruta a una red a la que no estoy directamente conectado es la siguiente:

```
route del -net {<red> netmask <máscara> | <red/prefijo>} [gw <puerta_de_enlace>] [<interfaz>]
```

Elimina una ruta estática a una red no directamente conectada a través de un equipo (gw). Si sólo hay una red a través de un equipo se puede omitir la opción `gw`.

### Eliminar ruta estática

Eliminar la ruta estática a la red de destino `192.168.11.0/25`.

```
root@Host:~# route del -net 192.168.11.0/25
```

Alternativamente, podemos utilizar el comando:

```
root@Host:~# route del -net 192.168.11.0 netmask 255.255.255.128
```

## PERSISTENCIA

Para que los cambios realizados en la tabla de enrutamiento sean **persistentes**, es decir, que se mantengan después de reiniciar el equipo, hay que añadir las rutas estáticas en los archivos de configuración de las interfaces de red. En el caso de las **distribuciones basadas en Debian**, disponemos de unas directivas que ejecutan un comando o script cuando se activa una interfaz y cuando se desactiva. Las directivas son las siguientes:

- `up <comando>`: el comando se ejecuta cuando se activa la conexión.
- `post-up <comando>`: el comando se ejecuta después de activar la conexión. Si la conexión no se activa correctamente, el comando no se ejecuta.
- `pre-down <comando>`: el comando se ejecuta antes de que se desactive la conexión.
- `down <comando>`: el comando se ejecuta cuando se desactiva la conexión.



Normalmente, lo que se hace con una instrucción `post-up` cuando se activa la conexión se debe deshacer con `pre-down` cuando se desactiva, aunque no es obligatorio.

Para añadir una ruta persistente solo tenemos que utilizar la directiva `post-up` (o `up`) con el comando `route` que añade la ruta y para eliminar una ruta utilizamos el comando `route` con la directiva `pre-down` (o `down`).

### Añadir y eliminar rutas persistentes

En una red tenemos un router que está directamente conectado a las redes `192.168.1.0/24` por la interfaz `eth0` y `192.168.2.0/25` por `eth1`. Necesita acceder a la red `192.168.2.128/25` a la cual se accede a través de la puerta de enlace `192.168.1.2`.

Configuramos las interfaces de red de manera que cuando se active la interfaz `eth0` se cree una ruta estática que da acceso a la red `192.168.2.128/25` y cuando se desactive se elimine la ruta.

```
auto eth0
iface eth0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    post-up route add -net 192.168.2.128/25 gw 192.168.1.2 dev eth0
    pre-down route del -net 192.168.2.128/25 gw 192.168.1.2 dev eth0

auto eth1
iface eth1 inet static
    address 192.168.2.1
    netmask 255.255.255.128
```

## TABLA ARP

El comando `arp` se utiliza para manipular o mostrar las entradas de la tabla ARP (Address Resolution Protocol), que almacena la correspondencia entre direcciones IP y direcciones MAC de los equipos en una red local.

## MOSTRAR TABLA

```
arp [-a | -v] [<ip_equipo>]
```

Muestra el contenido de la tabla ARP. Si se utiliza la opción `-a`, muestra la tabla en formato BSD. La opción `-v` proporciona información adicional (verbose). Si le pasamos una ip ( `<ip_equipo>` ) solo muestra las entradas correspondientes a ese equipo.

### Mostrar tabla ARP

Mostrar todas las entradas de la tabla ARP.

```
root@Host:~# arp
Address      Hwtype    Hwaddress  Flags      Mask      Iface
192.168.2.3  ether     08:00:27:3b:72:33  C          eth0
192.168.2.1  ether     52:54:00:12:35:00  C          eth0
```

# MODIFICAR TABLA

Se pueden añadir y eliminar entradas de la tabla ARP.

## AÑADIR ENTRADA

```
arp -s <ip_equipo> <mac>
```

Añade manualmente una entrada a la tabla ARP especificando la dirección IP del equipo y su dirección MAC.

### Añadir entrada ARP

Añadir una entrada a la tabla ARP para el equipo con IP `192.168.2.10` con mac `08:00:27:aa:bb:cc` y a continuación visualizar la tabla ARP.

```
root@Host:~# arp -s 192.168.2.10 08:00:27:aa:bb:cc
root@Host:~# arp
Address      Hwtype      Hwaddress    Flags        Mask         Iface
192.168.2.3  ether      08:00:27:3b:72:33  C              eth0
192.168.2.10 ether      08:00:27:aa:bb:cc  CM             eth0
192.168.2.1  ether      52:54:00:12:35:00  C              eth0
```

El valor de `Flags` a `M` indica que la entrada se ha añadido manualmente y no se ha aprendido dinámicamente.

## BORRAR ENTRADA

```
arp -d <ip_equipo>
```

Elimina la entrada de la tabla ARP correspondiente a la dirección IP especificada.

### Eliminar entrada ARP

Eliminar la entrada de la tabla ARP para el equipo con IP `192.168.2.10`.

```
root@Host:~# arp -d 192.168.2.10
```