

# CONFIGURACIÓN NO PERSISTENTE DE LA RED EN LINUX

 [Descargar PDF](#)

## ÍNDICE

- [CONFIGURACIÓN NO PERSISTENTE DE LA RED](#)
- ▼ [LA LIBRERÍA NET-TOOLS](#)
  - ▼ [IFCONFIG](#)
    - [MOSTRAR LAS INTERFACES DE RED](#)
    - [MOSTRAR ESTADÍSTICAS DE LAS INTERFACES DE RED](#)
    - [ACTIVAR O DESACTIVAR UNA INTERFAZ DE RED](#)
    - [CAMBIAR LA DIRECCIÓN MAC DE UNA INTERFAZ DE RED](#)
    - [CAMBIAR LA DIRECCIÓN IP DE UNA INTERFAZ DE RED](#)
  - ▼ [ROUTE](#)
    - [MOSTRAR LA TABLA DE ENRUTAMIENTO](#)
    - [AÑADIR UNA RUTA ESTÁTICA](#)
    - [ELIMINAR UNA RUTA ESTÁTICA](#)
  - ▼ [ARP](#)
    - [MOSTRAR LA TABLA ARP](#)
    - [AÑADIR UNA ENTRADA A LA TABLA ARP](#)
    - [ELIMINAR UNA ENTRADA DE LA TABLA ARP](#)
- ▼ [LA LIBRERÍA IPROUTE2](#)
  - ▼ [IP LINK](#)
    - [MOSTRAR EL ESTADO DE LAS INTERFACES DE RED](#)
    - [ACTIVAR Y DESACTIVAR INTERFACES DE RED](#)
    - [CAMBIAR LA MAC DE UNA INTERFAZ DE RED](#)
  - ▼ [IP ADDRESS](#)
    - [MOSTRAR LA CONFIGURACIÓN IP DE LAS INTERFACES DE RED](#)
    - [CONFIGURAR LAS IP DE LAS INTERFACES DE RED](#)
    - [ELIMINAR IP DE LAS INTERFACES DE RED](#)
  - ▼ [IP ROUTE](#)
    - [CONSULTAR TABLA DE ENRUTAMIENTO](#)
    - [AÑADIR RUTAS A LA TABLA DE ENRUTAMIENTO](#)
    - [ELIMINAR RUTAS DE LA TABLA DE ENRUTAMIENTO](#)
  - [IP NEIGHBOUR](#)
- [PRUEBAS DE CONECTIVIDAD](#)
- [Configuración dinámica de red](#)

# CONFIGURACIÓN NO PERSISTENTE DE LA RED

La **configuración de bajo nivel** o **no persistente** de red en Linux 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.

Para realizar configuraciones de red no persistentes, se utilizan herramientas como **net-tools** o **iproute2**. Estas herramientas permiten configurar interfaces de red, asignar direcciones IP, establecer rutas y gestionar la conectividad de red sin necesidad de editar archivos de configuración permanentes.

La librería **net-tools** es la más antigua y tradicional, mientras que **iproute2** es la más moderna y recomendada en las distribuciones actuales de Linux. Aunque net-tools aún está disponible, se considera obsoleta y se recomienda utilizar iproute2 para nuevas configuraciones.

## LA LIBRERÍA NET-TOOLS

La librería **net-tools** dispone de las utilidades clásicas para la configuración de red en Linux. Aunque es menos común en las distribuciones modernas, aún se puede utilizar si se instala el paquete **net-tools**. Esta librería incluye herramientas como `ifconfig`, `route` y `arp`, que permiten configurar y gestionar las interfaces de red de forma temporal.

### IFCONFIG

La utilidad o comando **ifconfig** se utiliza para configurar y mostrar información de las interfaces de red. Entre las diferentes operaciones que podemos llevar a cabo con el comando ifconfig tenemos:

#### MOSTRAR LAS INTERFACES DE RED

```
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.

## MOSTRAR ESTADISTICAS DE LAS INTERFACES DE RED

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

Muestra estadísticas de todas las interfaces (o una concreta) activas.

## ACTIVAR O DESACTIVAR UNA INTERFAZ DE RED

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

Para activar una interfaz de red se utiliza el flag `up` y para desactivarla el flag `down`.

### Activar una interfaz de red

Activar la interfaz `eth0`.

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

## CAMBIAR LA DIRECCIÓN MAC DE UNA INTERFAZ DE RED

```
ifconfig <interfaz> hw ether <mac>
```

Cambia la dirección MAC de una interfaz de red. Es importante tener en cuenta que al reiniciar el sistema, la interfaz recuperará su dirección MAC original.

### Cambiar dirección MAC

Cambiar la dirección MAC de la interfaz `eth0` a `00:00:2a:3a:4a:5a`.

```
root@Host:~# ifconfig eth0 hw ether 00:00:2a:3a:4a:5a
```

## CAMBIAR LA DIRECCIÓN IP DE UNA INTERFAZ DE RED

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

Cambia la dirección IP de una interfaz de red o le asigna una si no tenía. La configuración de la dirección IP con ifconfig es **estática** (o manual).

### Cambiar 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
```

## ROUTE

La utilidad o comando `route` se utiliza para configurar y mostrar información de la tabla de enrutamiento de un equipo; es decir la tabla que nos indica como llegar a otras redes a las que no se está directamente conectado y a Internet. Entre las diferentes operaciones que podemos llevar a cabo con el comando route tenemos:

### MOSTRAR LA TABLA DE ENRUTAMIENTO

```
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
```

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`.

```
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
```

## AÑADIR UNA RUTA ESTÁTICA

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.

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

Añadir una ruta por defecto (a Internet) a través de una puerta de enlace. La opción `dev` se puede omitir si sólo hay una interfaz de red; es más se puede omitir hasta el nombre de la interfaz.

### Añadir ruta por defecto

Añadir una ruta por defecto a través de la puerta de enlace `192.168.2.1`. Suponiendo que la única interfaz de red es `eth0`.

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

```
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 `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
```

## ELIMINAR UNA RUTA ESTÁTICA

De forma análoga a como se añadían rutas, podemos eliminar rutas.

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

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

### Eliminar ruta por defecto

Eliminar la ruta por defecto de la tabla de enrutamiento de un equipo.

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

```
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
```

## 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. Entre las diferentes operaciones que podemos llevar a cabo con el comando arp tenemos:

### MOSTRAR LA TABLA ARP

```
arp [-a | -v] [<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).

### 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
```

## AÑADIR UNA ENTRADA A LA TABLA ARP

```
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.

## ELIMINAR UNA ENTRADA DE LA TABLA ARP

```
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
```

## LA LIBRERÍA IPROUTE2

La librería **iproute2** es la más moderna y recomendada en las distribuciones actuales de Linux. Proporciona una herramienta unificada llamada `ip` para gestionar la red de forma avanzada. Esta herramienta reemplaza a las utilidades clásicas de **net-tools** y ofrece una interfaz más coherente y potente para la configuración de red.

Con el comando `ip` se pueden realizar tareas que antes requerían múltiples comandos separados, como `ifconfig`, `route` y `arp`. La sintaxis del comando `ip` es flexible y permite gestionar diversos aspectos de la configuración de red, incluyendo interfaces, direcciones IP, rutas y tablas ARP. La sintaxis básica del comando `ip` es la siguiente:

```
ip [OPTIONS] OBJECT {COMMAND | help}
```

Las opciones (**OPTIONS**) y el objeto (**OBJECT**) determinan qué aspecto de la configuración de red se va a gestionar, mientras que el comando (**COMMAND**) especifica la acción a realizar. La opción `help` proporciona

información sobre el uso del comando.

Los valores posibles para **OPTIONS**, **OBJECT** y **COMMAND** son numerosos. Los valores más comunes son de **OBJECT** son:

- **link**: Utilizado para la gestión de las interfaces de red a nivel de enlace.
- **address**: Utilizado para la gestión del direccionamiento de red.
- **route**: Utilizado para la gestión de las tablas de enrutamiento.
- **neighbour**: Utilizado para la gestión de las tablas ARP.

El comando `ip` admite abreviaciones de las diferentes opciones con que se ejecuta siempre y cuando no haya ambigüedad.

## IP LINK

El comando `ip` con el objeto **link** nos permite llevar a cabo acciones como: activar y desactivar interfaces de red, ver el estado de éstas y cambiarles la dirección MAC.

### MOSTRAR EL ESTADO DE LAS INTERFACES DE RED

```
ip link show [up] [dev] [<interfaz>]
```

Muestra información del estado de una interfaz (o de todas si no se especifica interfaz). Con el flag `up` solo muestra las interfaces activas. La palabra reservada `dev` la podemos omitir.

Una forma abreviada del comando para mostrar todas las interfaces de red sería la siguiente:

```
ip link
```

#### Mostrar estado interfaces de red

Mostrar el estado de todas las interfaces e red que estén activas.

```
root@Host:~# ip link show up
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:f2:b7:02 brd ff:ff:ff:ff:ff:ff
```

Entre la información mostrada podemos comprobar que tenemos dos interfaces de red: la interfaz de loopback (**lo**) y la de la tarjeta de red **enp0s3**. Además se puede observar que las dos interfaces están activas (flag `UP`) y que la MAC de la tarjeta de red es `08:00:27:f2:b7:02`.

#### Mostrar estado interfaces de red

Mostrar el estado de la interfaz de red **enp0s8**.

```
root@Host:~# ip link dev enp0s8
```

Alternativamente, podríamos haber utilizado el comando:

```
root@Host:~# ip link enp0s8
```

## ACTIVAR Y DESACTIVAR INTERFACES DE RED

```
root@Host:~# ip link set [dev] <interfaz> [up|down]
```

Activa una interfaz de red (opción `up`) y desactiva una interfaz de red (opción `down`).

### Activar y desactivar interfaces de red

Desactivar la interfaz de red **enp0s3**.

```
root@Host:~# ip link set dev enp0s3 down
```

## CAMBIAR LA MAC DE UNA INTERFAZ DE RED

```
root@Host:~# ip link set [dev] <interfaz> address <nueva-mac>
```

Cuando cambiamos la MAC de una interfaz de red con el comando `ip link set` este cambio es temporal y no persistente; es decir, se pierde al reiniciar el sistema.

### Cambiar MAC interfaz de red

Cambiar la MAC de la interfaz de red **enp0s3** por la MAC `00:00:2a:3a:4a:5a`.

```
root@Host:~# ip link set dev enp0s3 00:00:2a:3a:4a:5a
```

Si mostramos el estado de la interfaz de red con el comando `ip link show enp0s3` podemos comprobar que la MAC ha cambiado. También podemos observar que aparece el campo `permaddr` que nos indica la MAC original de la interfaz de red.

```
root@Host:~# ip link show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 00:00:2a:3a:4a:5a brd ff:ff:ff:ff:ff:ff permaddr 08:00:27:f2:b7:02
```

## IP ADDRESS

El comando `ip` con el objeto **address** nos permite llevar a cabo acciones como: mostrar la configuración de las interfaces de red y asignar direcciones IP a las mismas.

## MOSTRAR LA CONFIGURACIÓN IP DE LAS INTERFAZES DE RED

```
ip address show [up] [dev <interfaz>]
```

Muestra la configuración del direccionamiento de todas las interfaces de red o la de la interfaz pasada como parámetro. Con la opción `up` se muestran solo las que están activadas. La información más relevante que

muestra este comando es:

- El nombre de la interfaz (o interfaces) de red
- La dirección IPv4
- La dirección de broadcast
- La dirección MAC
- El estado de la interfaz. El flag `UP` y el indicador `state UP` indican que la interfaz está activa (si no aparecen no está activa).

## Mostrar configuración interfaces

Muestra la configuración de todas las interfaces de red, estén o no activadas.

```
root@Host:~# ip address show
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_fast2 state UP group default qlen 1000
    link/ether 08:00:27:f2:b7:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.17/24 brd 192.168.2.255 scope global dynamic enp0s3
        valid_lft 310sec preferred_lft 310sec
    inet6 fe80::a00:27ff:fe2:b702/64 scope link
        valid_lft forever preferred_lft forever
```

En la imagen se muestra la salida del comando `ip addr show` en un equipo con una única interfaz de red activa (**enp0s3**) y la interfaz de loopback (**lo**).

En la salida del comando se puede observar:

- El nombre de la interfaz de red (**enp0s3** y **lo**).
- La dirección MAC (para **enp0s3** sería `08:00:27:f2:b7:02` ).
- La dirección IPv4 (para **enp0s3** sería `192.168.1.10/24` ).
- La dirección de broadcast (para **enp0s3** sería `192.168.1.255` ).
- El estado de la interfaz. El flag `UP` y el indicador `state UP` indican que la interfaz está activa.

Cualquiera de los comandos mostrados a continuación son equivalentes, pues como ya sabemos se admiten abreviaturas.

```
root@Host:~# ip address
root@Host:~# ip addr show
root@Host:~# ip addr
root@Host:~# ip a sh
root@Host:~# ip a
```

## Mostrar configuración interfaces

Muestra la configuración de la interfaz **enp0s3**.

```
root@Host:~# ip address show dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:f2:b7:02 brd ff:ff:ff:ff:ff:ff
        inet 192.168.2.17/24 brd 192.168.2.255 scope global dynamic enp0s3
            valid_lft 283sec preferred_lft 283sec
        inet6 fe80::a00:27ff:fef2:b702/64 scope link
            valid_lft forever preferred_lft forever
```

Las opciones `UP` y `state UP` indican que la interfaz está activada. Si no aparecen es que no está activada.

### Mostrar configuración interfaces

Hemos desactivado la interfaz **enp0s3**. Primero mostramos la configuración de todas las interfaces y a continuación solo de las activas.

Mostrar todas las interfaces.

```
root@Host:~# ip address show
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> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 08:00:27:f2:b7:02 brd ff:ff:ff:ff:ff:ff
        inet 192.168.2.17/24 brd 192.168.2.255 scope global dynamic enp0s3
            valid_lft 225sec preferred_lft 225sec
```

Fijémonos que no aparece el indicador de estado `UP` en la interfaz **enp0s3**. Además, el indicador `state DOWN` nos dice que la interfaz no está activa.

Ahora mostramos solo las interfaces activas.

```
root@Host:~# ip address show up
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
```

Ahora solo se muestra la interfaz de loopback **lo** ya que la interfaz **enp0s3** está desactivada.

## CONFIGURAR LAS IP DE LAS INTERFACES DE RED

```
ip addr add <ip>/<prefijo> dev <interfaz>
```

Añade una nueva IP a una interfaz de red. Si ya tenía una, ahora tendrá una más.

### Añadir IP a interfaz de red

Añadir la IP `192.168.2.128` a la interfaz de red **enp0s3**.

```
root@PCL-1:~# ip address add 192.168.2.128/24 dev enp0s3
```

Podemos comprobar que ahora tenemos dos IPs:

```
root@Host:~# ip address show dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:f2:b7:02 brd ff:ff:ff:ff:ff:ff
        inet 192.168.2.17/24 brd 192.168.2.255 scope global dynamic enp0s3
            valid_lft 106sec preferred_lft 106sec
        inet 192.168.2.128/24 scope global secondary enp0s3
            valid_lft forever preferred_lft forever
```

## ELIMINAR IP DE LAS INTERFACES DE RED

```
ip address {delete | del} <ip/prefix> dev <interfaz>
```

Elimina la IP especificada de la interfaz de red dada.

### Eliminar IP de interfaces de red

Eliminar la IP **192.168.2.128** de la interfaz de red **enp0s3**.

```
root@Host:~# ip address delete 192.168.2.128/24 dev enp0s3
```

También habría sido válido hacer:

```
root@Host:~# ip address del 192.168.2.128/24 dev enp0s3
```

Es muy habitual escribir **addr** como forma abreviada de **address**:

```
root@Host:~# ip addr del 192.168.2.128/24 dev enp0s3
```

```
ip address flush [[dev] <interfaz>] [<opción>]
```

Elimina todas las IPs asignadas a una interfaz dada (o a todas) que cumplen una propiedad especificada por el parámetro **opción**. Algunas de las opciones son:

- **dynamic** : para IPs asignadas por un servidor DHCP,
- **permanent** : para IPs configuradas manualmente,
- **secondary** : para IPs secundarias,
- **up** : para direcciones de interfaces activas que tengan IP asignada.

Con la tecla **Tab** antes de finalizar de escribir la instrucción podemos ver todas las opciones disponibles.

### Eliminar IP de interfaces de red

Elimina todas las IPs asignadas a la interfaz **enp0s3**.

```
root@Host:~# ip address flush dev enp0s3
```

### Eliminar IP de interfaces de red

Elimina todas las IPs que se han asignado dinámicamente a cualquier interfaz.

```
root@Host:~# ip address flush dynamic
```

### Eliminar IP de interfaces de red

Elimina todas las IPs que se han asignado de manera estática a la interfaz **enp0s8**.

```
root@Host:~# ip address flush dev enp0s8 permanent
```

## IP ROUTE

Utilizado para consultar y modificar la tabla de enrutamiento.

### CONSULTAR TABLA DE ENRUTAMIENTO

```
ip route [show]
```

Muestra la tabla de enrutamiento. No hay diferencia entre usar el parámetro `show` o no. Se muestra una línea por cada red que se puede alcanzar. Entre los datos más significativos se incluyen:

- La red de destino a la cual se deben enviar los datos.
- Para aquellas redes a las que no se está directamente conectado, se indica la puerta de enlace o IP del equipo al cual debemos enviar los datos para que lleguen a su destino.
- Interfaz por donde se deben enviar los datos. Si solo tenemos una tarjeta de red, siempre usaremos la misma interfaz.
- Para redes directamente conectadas siempre aparece el valor `scope link`.
- Para aquellas redes a las que no se está directamente conectado, la IP por la cual me conecto a dicha red.

La entrada de la tabla con red de destino `default` (`0.0.0.0/0`) hace referencia a la ruta por defecto; es decir donde se envían los paquetes si no hay ninguna entrada para la red de destino en la tabla de enrutamiento. Es importante tener una ruta por defecto ya que en caso contrario, seguramente, no tendremos acceso a Internet.

### Mostrar tabla de enrutamiento

Muestra la tabla de enrutamiento.

```
root@Host:~# ip route show
default via 192.168.2.1 dev enp0s3
192.168.2.0/24 dev enp0s3 proto kernel scope link src 192.168.2.17
```

La primera línea de la tabla de enrutamiento nos indica que tenemos definida una ruta por defecto y que en este caso, los paquetes se dirigen a la ip 192.168.2.1 a través de la interfaz enp0s3.

La segunda línea nos indica que la interfaz enp0s3 tiene la ip 192.168.2.17 y está directamente conectada a la red 192.168.2.0/24.

## AÑADIR RUTAS A LA TABLA DE ENRUTAMIENTO

Con ip route podemos añadir rutas para acceder a redes a las que no estoy directamente conectado. De especial importancia es la red por defecto que nos sirve para acceder a Internet u otras redes para las que no he definido ninguna ruta.

```
ip route add default via <puerta-enlace> [dev <interfaz>]
```

Añade una ruta por defecto de manera que los datos que no sean para ninguna de las otras redes de la tabla de encaminamiento se envíen a la puerta de enlace dada y salgan por la interfaz dada. Si solo tenemos una interfaz de red, podemos omitir `dev <interfaz>`.

### Añadir entrada a la tabla de enrutamiento

Configuración puerta de enlace de un equipo que solo tiene una tarjeta de red si sabemos que la puerta de enlace tiene la IP `192.168.2.1`.

```
root@Host:~# ip route add default via 192.168.2.1
```

### Añadir ruta por defecto a la tabla de enrutamiento

Configuración de la puerta de enlace de un equipo que tiene dos tarjetas de red `eth0` y `eth1`, de manera que la primera tarjeta (`enp0s3`) está en la red interna `192.168.254.0/24` y la segunda (`enp0s8`) en la red `192.168.1.0/24`, donde hay un enrutador que da acceso a internet cuya IP es la primera de la red.

```
root@Host:~# ip route add default via 192.168.1.1 dev enp0s8
```

```
ip route add <red-destino/prefix-red> via <puerta-enlace> [dev <interfaz>]
```

Añade una ruta estática para llegar a una red a la que no se está directamente conectado. Se debe indicar cuál es la red a la que queremos acceder (`<red-destino/prefix-red>`), por qué interfaz (`<interfaz>`) deben salir los datos y a qué equipo (`<puerta-enlace>`) debemos enviar los datos para que lleguen a su destino.

### Añadir ruta estática a la tabla de enrutamiento

Añadir una ruta estática para llegar a la red `192.168.11.128/25` desde la red `192.168.2.0/24`, utilizando como puerta de enlace el equipo con IP `192.168.2.1` y como interfaz de salida `enp0s3`.

```
root@Host:~# ip route add 192.168.11.128/25 via 192.168.2.1 dev enp0s3
```

Si solo tenemos una interfaz de red, podemos omitir `dev <interfaz>`.

Podemos ver que se ha añadido la nueva ruta:

```
root@Host:~# ip route show
default via 192.168.2.1 dev enp0s3
192.168.2.0/24 dev enp0s3 proto kernel scope link src 192.168.2.17
192.168.11.128/25 via 192.168.2.1 dev enp0s3
```

## ELIMINAR RUTAS DE LA TABLA DE ENRUTAMIENTO

De la misma manera que se pueden añadir rutas a la tabla de enrutamiento se pueden eliminar las rutas añadidas. También es posible modificar rutas con `ip route change` o `ip route replace` aunque esto no lo trataremos.

```
ip route delete default [via <puerta-enlace>]
```

Elimina la ruta por defecto dada la puerta de enlace. Si solo tenemos una ruta por defecto, no es necesario especificar la puerta de enlace (es lo habitual).



### Eliminar ruta por defecto de la tabla de enrutamiento

Eliminar la ruta por defecto si solo hay una.

```
root@Host:~# ip route delete default
```

Se suele utilizar la abreviatura `del`:

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

```
ip route delete <red-destino/prefix-red> [via <puerta-enlace>]
```

Elimina la ruta estática a la red especificada. Si solo hay una ruta estática con destino la red especificada, no es necesario indicar la puerta de enlace; en caso contrario, sí se debe especificar.



### Añadir ruta estática a la tabla de enrutamiento

Elimina la ruta estática con destino la red `192.168.11.128/25` si a esta se llega por el equipo con IP `192.168.2.1`. Sabemos que hay más de una ruta con este destino.

```
root@Host:~# ip route delete 192.168.11.128/25 via 192.168.2.1
```

# IP NEIGHBOUR

## PRUEBAS DE CONECTIVIDAD

Para concluir trataremos el tema de la conectividad. Cuando se configura el direccionamiento de red de los equipos, debemos garantizar que existe conectividad interna, conectividad externa con internet u otras redes a las que no estoy directamente conectado, y que funcione la resolución de nombres.

Para comprobar la conectividad podemos utilizar el comando `ping`. El comando `ping` envía un paquete ICMP de petición de eco a un equipo y este nos envía de respuesta otro paquete ICMP llamado de respuesta de eco.

La mejor manera de comprobar la conectividad interna es hacer un `ping` a la puerta de enlace o a otro equipo de la red que normalmente está en funcionamiento. Para la conectividad externa se puede hacer un `ping` a una dirección IP pública conocida, como la del servidor DNS de Google con IP `8.8.8.8`. Para comprobar la resolución DNS (necesaria, por ejemplo, cuando usamos un navegador para ir a una página web), debemos tener correctamente configurados los clientes DNS. Estos se configuran de manera permanente en un archivo del sistema.

### Conectividad interna

Comprobamos que tenemos conectividad interna desde un equipo de la red `192.168.2.0/24` haciendo un `ping` a la puerta de enlace cuya IP es `192.168.2.1`.

```
root@Host:~# ping -c2 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=255 time=0.361 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=255 time=0.391 ms

--- 192.168.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1024ms
rtt min/avg/max/mdev = 0.361/0.376/0.391/0.015 ms
```

Vemos que sí hay conectividad ya que se han enviado dos paquetes y no se ha perdido ninguna de las respuestas. El parámetro `-c2` indica que solo envía dos paquetes (por defecto se envían paquetes hasta que el usuario pulse Ctrl+C).

Si no nos llega ninguna respuesta, probablemente el problema es que no tenemos configurada la ruta por defecto. Esto lo podemos comprobar con el comando `ip route`.

### Conectividad externa

Para comprobar la conectividad externa podemos hacer un `ping` a la IP `8.8.8.8`.

```
root@Host:~# ping -c2 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=13.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=12.8 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 12.782/12.924/13.067/0.142 ms
```

Si la puerta de enlace de la ruta por defecto está bien configurada, esta comprobación debería funcionar.

El hecho de que el `ping` no funcione no significa necesariamente que la configuración de la red esté mal hecha. Puede haber un cortafuegos que filtre las peticiones de eco. Por ejemplo, el cortafuegos Windows Defender no permite, por defecto, las peticiones de eco.

En estos casos podemos utilizar aplicaciones alternativas para comprobar la conectividad. Por ejemplo, la aplicación `hping3` permite enviar paquetes de petición de eco de muchos protocolos, no solo ICMP.

## Conectividad externa

Hacemos un `ping` convencional a uno de los servidores web de la Generalitat Valenciana cuya IP es `195.77.16.26`. Cuando hacemos un `ping` vemos que el equipo destino o un equipo intermedio bloquea las peticiones de eco.

```
root@PCL-1:~# ping -c2 195.77.16.26
PING 195.77.16.26 (195.77.16.26) 56(84) bytes of data.

--- 195.77.16.26 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1029ms
```

No nos ha llegado ninguna respuesta, como se puede ver.

Instalamos el paquete `hping3`:

```
root@PCL-1:~# apt install hping3
```

Hacemos un `hping3` a la IP del servidor web de la Generalitat, enviando un paquete de conexión HTTP, y vemos que sí tenemos respuesta:

```
root@PCL-1:~# hping3 -c 2 -S -p 80 195.77.16.26
HPING 195.77.16.26 (enp0s3 195.77.16.26): S set, 40 headers + 0 data bytes
len=46 ip=195.77.16.26 ttl=255 id=42 sport=80 flags=SA seq=0 win=32768 rtt=19.8 ms
len=46 ip=195.77.16.26 ttl=255 id=43 sport=80 flags=SA seq=1 win=32768 rtt=19.7 ms

--- 195.77.16.26 hping statistic ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 19.7/19.8/19.8 ms
```

Podemos encontrar más información de la utilidad `hping3` en las páginas de manual:

```
root@PCL-1:~# man hping3
```

Para conocer la sintaxis del comando, podemos acceder a la ayuda contextual con:

```
root@PCL-1:~# hping3 --help
```

En los sistemas Alpine Linux, el paquete `hping3` no se encuentra en los repositorios del sistema, sino en un repositorio de prueba. Podemos descargarlo e instalarlo de este repositorio con el comando:

```
AlpineLinux:~# apk add hping3 --update-cache --repository http://dl-cdn.alpinelinux.org/alpine/edge/testing
fetch http://dl-cdn.alpinelinux.org/alpine/v3.18/testing/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.18/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/edge/testing/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.18/main/x86_64/APKINDEX.tar.gz
(1/4) Installing libpcap (1.10.4-r1)
(2/4) Installing tzdata (2024a-r0)
(3/4) Installing tcl (8.6.13-r1)
(4/4) Installing hping3 (20051105-r4)
Executing busybox-1.36.1-r2.trigger
OK: 160 MiB in 71 packages
```

```
AlpineLinux:~#
```

## Configuración dinámica de red

Para configurar una interfaz de red de forma dinámica, es decir, para que la dirección IP sea asignada automáticamente por un servidor DHCP, es necesario utilizar el cliente DHCP que proporciona la distribución de Linux.

En Debian y sus derivados, el cliente DHCP es **dhclient**.

- **Solicitar una dirección IP por DHCP**

Para solicitar una dirección IP de forma dinámica a través de un servidor DHCP, se utiliza el cliente DHCP proporcionado por la distribución de Linux. En Debian y sus derivados, el cliente DHCP es **dhclient**.

```
sudo dhclient <interfaz>
```

Solicita una dirección IP para la interfaz especificada desde un servidor DHCP.

### Solicitar IP por DHCP

Solicitar una dirección IP para la interfaz `eth0` utilizando el cliente DHCP.

```
sudo dhclient eth0
```

```
sudo dhclient enp0s3
```

Al igual que con net-tools, estos cambios son **temporales** y se perderán tras un reinicio.