

# Gestión de red por comandos con iproute2 (comando ip) Guía de Linux

8–10 minutos

Las net-tools se marcaron hace bastante tiempo como obsoletas, fue entonces cuando la suite de herramientas iproute2 que se encuentran más actualizadas y mejoran algunos aspectos además de añadir funciones que antes no podían hacerse han tomado por sorpresa a algunos usuarios que aún no se habían decidido a adaptarse.

Ha sido recientemente tras ir actualizando distribuciones a nuevas versiones han dejado de traer este software preinstalado por defecto y es cuando nos hemos visto obligados a utilizar las iproute2 en vez de las net-tools y con ello abandonar nuestro querido ifconfig que tanto tiempo nos ha acompañado.

## Gestión de Red con iproute2



- [Mostrar todos los dispositivos de red](#)
- [Mostrar solo detalles de una interfaz de red](#)
- [Añadir dirección IP](#)
- [Agregar puerta de Enlace](#)
- [Eliminar puerta de enlace por defecto](#)
- [Eliminar dirección IP de una interfaz](#)
- [Limpiar configuración de red para una interfaz](#)
- [Añadir interfaz virtual \(alias\)](#)
- [Eliminar interfaz virtual](#)
- [Ver tablas de rutas](#)
- [Añadir ruta](#)
- [Eliminar ruta](#)
- [Ver tabla ARP](#)
- [Añadir entrada a la tabla ARP](#)
- [Desconectar Dispositivo ARP](#)
- [Modo Promisc](#)
- [Cambiar MTU de una interfaz](#)
- [Cambiar MAC temporalmente](#)
- [Cambiar MAC automáticamente con systemd](#)
- [Cambiar MAC con regla udev](#)

## Gestión Básica de la red

Listados de comandos básicos para establecer los parámetros necesarios de una conexión de red como puede ser la propia dirección, máscara de difusión, puerta de enlace por defecto...

Mostrar todos los dispositivos de red:

Hay distintas formas para mostrar las interfaces de red. Podéis probarlas ya que muestran de forma distinta los detalles cada una de ellas, elegid la que más os guste:

```
sudo ip a
```

```
sudo ip addr
```

```
sudo ip addr show
```

```
sudo ip addr list
```

```
sudo ip link show
```

Mostrar solo detalles de una interfaz de red

Sintaxis para mostrar los detalles de una interfaz, en los siguientes ejemplos usaré eth3

```
sudo ip a show eth3
```

```
sudo ip addr show dev eth3
```

```
sudo ip link show eth3
```

## Añadir dirección IP

Para añadir una dirección IP a una interfaz de red tenemos varias formas, a continuación dos de ellas:

```
sudo ip address add 192.168.1.1 dev eth3
```

```
sudo ip addr add 10.0.0.100/24 broadcast 10.0.0.255 dev eth3
```

## Agregar puerta de Enlace

La puerta de enlace será la salida de la red en la que nos encontramos. Para ello mostraré 2 ejemplos, el primero asignará la ruta de enlace para todas las interfaces y la segunda para una en concreto, en este caso el ejemplo continuará con la eth3.

```
sudo ip route add default via 192.168.1.1
```

```
sudo ip route add default via 192.168.1.1 dev eth3
```

## Eliminar puerta de enlace por defecto

El siguiente comando permite eliminar una puerta de enlace por defecto

```
sudo ip del default
```

## Eliminar dirección IP de una interfaz

Esta característica existe en esta suite de herramientas, a diferencia de las net-tools.

El siguiente ejemplo permite eliminar la dirección IP de una interfaz de red

```
sudo ip address del 192.168.1.1 dev eth3
```

Limpiar configuración de red para una interfaz

Este comando que mostraré a continuación vaciará de datos la interfaz de red dejando sin parámetros de conexión a la misma.

```
sudo ip address flush dev eth3
```

### Interfaces Virtuales (Alias)

Las interfaces virtuales son alias que permiten crear nuevas interfaces de red por software a partir de una interfaz real, de este modo podemos por ejemplo asignar varias direcciones IP al mismo dispositivo, pero con distintas interfaces (la real y la/las virtualizada/s).

### Añadir interfaz virtual (alias)

Os muestro dos formas para añadir una interfaz virtual.

```
sudo ip addr add 10.0.0.1/8 dev eth0 label eth3:1
```

```
sudo ip addr add 10.0.0.101/24 broadcast 10.0.0.255 dev eth3:1
```

## Eliminar interfaz virtual

Cuando no necesitemos la interfaz virtual que creamos antes podemos eliminarla con la siguiente orden.

```
sudo ip link set dev eth3:1 down
```

## Rutas y Tablas ARP

Las rutas y tablas ARP son los caminos a tomar hacia una red y las traducciones de equipos/MAC respectivamente. Es útil en muchas ocasiones tener acceso a estos ya sea para indicar como llegar a cierta red o para conocer las direcciones y MAC de dispositivos con los que hemos interactuados.

### Ver tablas de rutas

Estos parámetros son para mostrar las tablas de rutas actuales:

```
sudo ip route
```

```
sudo ip route show
```

### Añadir ruta

Podemos añadir rutas manualmente de la siguiente forma:

```
sudo ip route add 192.168.2.0/24 via 192.168.1.10
```

### Eliminar ruta

Igualmente que podemos añadir rutas, también podemos eliminar estas rutas así:

```
sudo ip route del 192.168.2.0/24
```

Ver tabla ARP

Con el siguiente comando podremos ver la tabla ARP con caché de una interfaz y en segundo lugar de todas las interfaces:

```
ip neighbor show dev eth3
```

```
ip neighbor show
```

Añadir entrada a la tabla ARP

La tabla ARP también se puede modificar manualmente (o vigilar para detectar intrusiones en la red o ataques) para añadir una entrada lo haremos de la siguiente forma:

```
sudo ip neigh add 192.168.0.1 lladdr 00:11:22:33:44:55 nud  
permanent dev eth3
```

Desconectar Dispositivo ARP

Para desactivar un dispositivo sus tablas ARP usaremos el siguiente comando:

```
ip link set dev eth0 arp off
```

## Otros Ajustes

A continuación algunos ajustes más que aunque es poco probable que necesites modificarlo puede ser interesante conocerlos para llegado el momento de necesitarse.

### Modo Promisc

El modo promiscuo trata de capturar paquetes que normalmente se descartarían ya sea porque no tenga datos relevantes o esté destinado a otra máquina.

Este modo es interesante en cuanto a la hora de analizar una red.

A continuación mostraré como activar y desactivar una interfaz de red en Modo Promiscuo:

```
sudo ip link set dev eth3 promisc on
```

```
sudo ip link set dev eth3 promisc off
```

### Cambiar MTU de una interfaz

El MTU (Maximum Transmission Unit) es un parámetro que indica la unidad máxima de transferencia usando un protocolo de comunicaciones. Podemos tener más información [aquí](#).

La mayoría de las redes en area local ethernet utilizan un MTU de 1500 bytes por defecto.

```
sudo ip link set dev eth3 mtu 9000
```



## Cambiar MAC

La MAC es un identificador único de cada dispositivo, pero aún así este puede ser cambiado por otro definido por nosotros. [aquí](#).

En realidad en ningún momento cambiará realmente, es decir, siempre tendrá el mismo identificador pero podemos decirle a la interfaz de red que le diga a la red que ese identificador es otro.

### Cambiar MAC temporalmente

Para cambiar la MAC temporalmente hasta que la devolvamos manualmente a su origen, reiniciemos el servicio de red o reiniciemos el sistema tenemos que ejecutar los siguientes 3 comandos en el mismo orden, cambiando la dirección MAC del ejemplo (XX:XX:XX:XX:XX:XX) por la que deseamos asignarle.

```
sudo ip link set dev eth3 interface down
```

```
sudo ip link set dev eth3 interface address XX:XX:XX:XX:XX:XX
```

```
sudo ip link set dev eth3 interface up
```

### Cambiar MAC automáticamente con systemd

Podemos establecer que la dirección MAC cambie automáticamente cuando se inicia systemd.

Esto lo llevaremos a cabo editando el archivo `/etc/systemd/network/00-default.link`

```
sudo /etc/systemd/network/00-default.link
```

Una vez dentro añadimos el siguiente contenido modificando los parámetros de nuestra interfaz elegida:

```
[Match]
```

```
MACAddress=original MAC
```

```
[Link]
```

```
MACAddress=spoofed MAC
```

```
NamePolicy=kernel database onboard slot path
```

Cambiar MAC con regla udev

Otro método posible es editar una regla udev, o crearla si no existiese.

Esta iría en el archivo `/etc/udev/rules.d/75-mac-spoof.rules`

La editaremos con nano por ejemplo:

```
sudo nano /etc/udev/rules.d/75-mac-spoof.rules
```

Y dentro editamos los parámetros necesarios para nuestra interfaz reemplazando X por nuestra dirección MAC real/original y también Y por la nueva dirección MAC.

```
ACTION=="add", SUBSYSTEM=="net",
```

```
ATTR{address}=="XX:XX:XX:XX:XX:XX", RUN+="/usr/bin/ip link set  
dev %k address YY:YY:YY:YY:YY:YY"
```