

# LINUX MODERNO

 [Descargar PDF](#)

## ▼ LINUX MODERNO

- INTRODUCCIÓN
- MOSTRAR INTERFACES DE RED
- MOSTRAR CONFIGURACIÓN INTERFACES DE RED
- MOSTRAR CONFIGURACIÓN DNS
- DESACTIVAR Y ACTIVAR INTERFACES
- CAMBIAR MAC A UNA INTERFAZ
- ▼ CONFIGURACIÓN ESTÁTICA
  - CONFIGURAR IP Y GATEWAY
  - BORRAR IP Y GATEWAY
- CONFIGURACIÓN DINÁMICA
- ▼ TABLA DE ENRUTAMIENTO
  - MOSTRAR TABLA
  - ▼ MODIFICAR TABLA
    - AÑADIR ENTRADA
    - ELIMINAR ENTRADA
- ▼ TABLA ARP
  - MOSTRAR TABLA
  - ▼ MODIFICAR TABLA
    - AÑADIR ENTRADA
    - ELIMINAR ENTRADA

# INTRODUCCIÓN

En el presente tema vamos a estudiar la configuración de la red en los sistemas Linux modernos; los basados en *systemd*. 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.

Al igual que en los sistemas tradicionales, distinguiremos dos formas diferentes de configuración de la red:

- **Configuración no persistente** o de bajo nivel. Para esta configuración se utilizan utilidades específicas incluidas en 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.

- **Configuración persistente**. Básicamente, hay dos formas de configurar persistentemente la red: mediante la aplicación *NetworkManager* y el servicio *systemd-networkd*.
  - **NetworkManager**: Inicialmente utilizado en los sistemas basados en Red Hat posteriormente se extendió a otras distribuciones Debian y derivadas, como Ubuntu. El objetivo era configurar la red sin modificar directamente los archivos de configuración utilizando los comandos y/o aplicaciones proporcionadas por *NetworkManager*. En las últimas versiones de Debian y con Ubuntu con entorno de escritorio *NetworkManager* está instalado y activado por defecto. A pesar de esto, todavía se puede configurar la red utilizando el sistema basado en el archivo `/etc/network/interfaces`, pero hay que tener cuidado de no utilizar los dos sistemas simultáneamente; es mejor deshabilitar *NetworkManager* en este caso.
  - **systemd-networkd**: Es un servicio de *systemd* que proporciona una forma de gestionar la configuración de red en sistemas Linux. A diferencia de *NetworkManager*, que está diseñado para entornos de escritorio, *systemd-networkd* está más orientado a servidores y sistemas sin entorno de escritorio. Utiliza archivos de configuración ubicados en el directorio `/etc/systemd/network/` para definir la configuración de red de forma declarativa.

A partir de la versión 17.10 en Ubuntu se incorpora una nueva herramienta llamada **Netplan** para administrar y configurar fácilmente la red en un archivo de configuración en formato *YAML*. *Netplan* es un

front-end que funciona conjuntamente con *NetworkManager* y con *systemd-networkd*. Configuramos los adaptadores con *Netplan* de una única manera independientemente de que el servicio de red sea *NetworkManager* o *systemd-networkd*. Solo tenemos que indicar cuál de los dos servicios de red estamos utilizando y *Netplan* hará todos los ajustes necesarios.

La configuración persistente de la red con *NetworkManager*, *systemd-networkd* y *Netplan* la desarrollaremos en otros documentos.

## MOSTRAR INTERFACES DE RED

Para este propósito utilizamos el comando `ip` con el objeto **link**. La sintaxis es la siguiente:

```
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
```

# MOSTRAR CONFIGURACIÓN INTERFACES DE RED

Para este propósito también utilizamos el comando `ip` pero con el objeto **address**. La sintaxis es la siguiente:

```
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 192.168.2.17/245 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.

# MOSTRAR CONFIGURACIÓN DNS

La configuración del DNS se realiza de forma persistente y por lo tanto mostrar su configuración depende del modelo de configuración persistente de la red utilizado.

# DESACTIVAR Y ACTIVAR INTERFACES

Utilizamos el comando `ip` con el objeto **link**. La sintaxis es la siguiente:

```
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` y activar la interfaz `enp0s8`.

```
root@Host:~# ip link set dev enp0s3 down
root@Host:~# ip link set dev enp0s8 up
```

# CAMBIAR MAC A UNA INTERFAZ

Cambiamos la MAC de una interfaz de red con el comando `ip link set`. La sintaxis es la siguiente:

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

Hay que tener en cuenta que este cambio es temporal y no persistente; es decir, se pierde al reiniciar el sistema y se recupera la MAC cableada en la tarjeta.

## 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
```

# CONFIGURACIÓN ESTÁTICA

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

Con el comando `ip` con el objeto **route** podemos gestionar todo lo relativo al gateway.

La configuración del **DNS** se realiza de **forma persistente** y por lo tanto depende del modelo de configuración persistente de la red utilizado.

## CONFIGURAR IP Y GATEWAY

Con el comando `ip address add` podemos asignar ip's a las interfaces de red. La sintaxis es la siguiente:

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

La ejecución del comando añade una nueva IP a una interfaz de red. Si ya tenía una, ahora tendrá una más.

Es más habitual utilizar las versiones abreviadas:

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

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



### Añadir IP a interfaz de red

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

```
root@Host:~# 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
```

Para añadir la puerta de enlace tenemos que añadir una ruta por defecto con el comando `ip route`. Su sintaxis es la siguiente:

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

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



### Configurar gateway

Configuración de la puerta de enlace (gateway) de un equipo que tiene dos tarjetas de red `enp0s3` y `enp0s8`, 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 router que da acceso a internet cuya IP es la primera de la segunda red.

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

# BORRAR IP Y GATEWAY

Para eliminar una **ip de una interfaz de red** tenemos el comando `ip address delete` . Su sintaxis es la siguiente:

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

## 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
```

También se pueden eliminar todas las ip's asignadas a una interfaz de red (una interfaz puede tener más de una ip) con el comando `ip address flush` .

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

La ejecución del comando elimina todas las IPs asignadas a una interfaz dada (o a todas) que cumplen una propiedad especificada por el parámetro `<opción>` . Algunos de los valores 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

Eliminar 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 de manera estática a la interfaz `enp0s8` .

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



Para eliminar una ruta por defecto (**gateway**) utilizamos el comando `ip route delete`. Si solo tenemos una ruta por defecto, no es necesario especificar la puerta de enlace (es lo habitual). Su sintaxis es la siguiente:

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

### Eliminar gateway

Eliminar la ruta por defecto (gateway) del equipo. Solo hay una ruta por defecto.

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

O utilizando la abreviatura `del`:

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

## CONFIGURACIÓN DINÁMICA

Para la configuración dinámica de la red utilizamos un cliente DHCP. El cliente DHCP es un programa que se ejecuta en el equipo y que se encarga de solicitar una dirección IP y otros parámetros de red a un servidor DHCP de la red. Este cliente depende de la distribución de Linux utilizada. Un cliente muy utilizado en las distribuciones basadas en Debian es `dhclient` visto en el documento de configuración tradicional de la red en Linux.

Para la liberación de la IP asignada dinámicamente además del cliente DHCP también podemos utilizar el comando `ip flush` con la opción *dynamic*.

### Liberar IP dinámicamente

Liberar la IP asignada dinámicamente con el cliente `dhclient` y con `ip flush` a la interfaz `enp0s3`.

Con `dhclient`:

```
root@Host:~# dhclient -r enp0s3
```

Con `ip flush`:

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

## TABLA DE ENRUTAMIENTO

La utilidad o comando `ip 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 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

La sintaxis para mostrar la tabla de enrutamiento es la siguiente:

```
ip route [show]
```

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
192.168.3.0/24 via 192.168.3.1 dev enp0s8
```

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`.

La tercera línea nos indica que la red `192.168.3.0/24` se alcanza a través de la puerta de enlace `192.168.3.1` y la interfaz `enp0s8`.

## MODIFICAR TABLA

También utilizaremos el comando `ip 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

Con `ip route add` 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. Este último caso ya fue tratado en la sección de la configuración del gateway. La sintaxis del comando es la siguiente:

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

Se debe indicar cuál es la red a la que queremos acceder ( `<red-destino/prefijo-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. Si solo tenemos una interfaz de red, podemos omitir `dev <interfaz>` .

### Añadir ruta estática

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
```

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 ENTRADA

Para eliminar las rutas añadidas se utiliza el comando `ip route delete` . También es posible modificar rutas con `ip route change` o `ip route replace` aunque esto no lo trataremos ya que es preferible eliminar la ruta y a continuación añadirla con los nuevos cambios. La sintaxis del comando es la siguiente:

```
ip route delete <red-destino/prefijo-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.

### Eliminar ruta estática

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
```

## TABLA ARP

Para gestionar las tablas ARP se utiliza el comando `ip` con el objeto **neighbour**. Este comando tiene una funcionalidad análoga a la del comando `arp` de la librería **net-tools**.

En la ayuda contextual se puede obtener más información sobre los estados de una entrada ARP.

## MOSTRAR TABLA

La sintaxis del comando en este caso es la siguiente:

```
ip neighbour [-s] [show [[<ip-equip>]]]
```

Muestra el contenido de toda la tabla ARP o solo la correspondiente a un equipo concreto. La última columna indica el estado de la entrada. Se puede obtener más información sobre los estados de una entrada ARP en la ayuda del comando ( `man ip` o `ip neighbour -h` ). Cuando pasamos la opción `-s` nos aporta información estadística.

Como habíamos comentado, se pueden utilizar abreviaturas en el comando. En este caso es habitual ejecutar el comando como: `ip neigh` o también `ip n`.

## Mostrar tabla ARP

Muestra todas las entradas de la tabla ARP.

```
root@Host:~# ip neigh
192.168.2.16 dev enp0s3 lladdr 08:00:27:d0:2f:54 REACHABLE
192.168.2.1 dev enp0s3 lladdr 52:54:00:12:35:00 REACHABLE
192.168.2.2 dev enp0s3 lladdr 52:54:00:12:35:00 REACHABLE
192.168.2.3 dev enp0s3 lladdr 08:00:27:3b:72:33 REACHABLE
```

Estas formas del comando serían todas válidas:

```
root@Host:~# ip neighbour show
root@Host:~# ip neighbour
root@Host:~# ip neigh
root@Host:~# ip neigh show
```

## Mostrar tabla ARP

Muestra la entrada de la tabla ARP correspondiente a la IP `192.168.2.16`.

```
root@PHost:~# ip neighbour show 192.168.2.16
192.168.2.16 dev enp0s3 lladdr 08:00:27:d0:2f:54 REACHABLE
```

Es posible mostrar las entradas ARP correspondientes a una interfaz concreta. En este caso el comando adopta la siguiente sintaxis:

```
ip neighbour show dev <interfaz>
```

Con `ip neighbour get` se pueden mostrar las entradas de la tabla ARP correspondientes a una interfaz y una IP. En este caso la sintaxis es:

```
ip neighbour get <ip-equipo> dev <interfaz>
```

También se pueden mostrar todas las entradas que estén en estado concreto con la opción `nud` del comando `ip neighbour show`. La sintaxis es la siguiente:

```
ip neighbour show nud <estado>
```

## Mostrar tabla ARP

Muestra todas las entradas que están en el estado llamado `stale`.

En un momento dado tenemos la siguiente tabla ARP:

```
root@Host:~# ip neighbour show
192.168.2.16 dev enp0s3 lladdr 08:00:27:d0:2f:54 STALE
192.168.2.1 dev enp0s3 lladdr 52:54:00:12:35:00 STALE
192.168.2.2 dev enp0s3 lladdr 52:54:00:12:35:00 STALE
192.168.2.3 dev enp0s3 lladdr 08:00:27:3b:72:33 REACHABLE
```

La última columna indica el estado de la entrada. El comando solicitado sería:

```
root@Host:~# ip neighbour show nud stale
192.168.2.16 dev enp0s3 lladdr 08:00:27:d0:2f:54 STALE
192.168.2.1 dev enp0s3 lladdr 52:54:00:12:35:00 STALE
192.168.2.2 dev enp0s3 lladdr 52:54:00:12:35:00 STALE
```

## MODIFICAR TABLA

Contemplaremos como añadir entradas a la tabla ARP y como eliminarlas. Es posible cambiar las entradas de la tabla ARP aunque no lo consideraremos.

### AÑADIR ENTRADA

Con el comando `ip neighbour add` añadimos entradas a la tabla ARP. Su sintaxis es la siguiente:

```
ip neighbour add <ip-equip> lladdr <mac> dev <interfaz>
```

Añade una entrada manualmente a la tabla ARP indicando la IP del equipo, su MAC y la interfaz por la que se llega al equipo con esta MAC.



#### Añadir entrada a tabla ARP

Añadir a la tabla ARP el equipo de IP `192.168.2.10` con MAC `08:00:27:ee:11:22` a la interfaz `enp0s3`.

```
root@Host:~# ip neigh add 192.168.2.10 lladdr 08:00:27:ee:11:22 dev enp0s3
```

Hemos utilizado la forma abreviada.

Podemos visualizar la entrada añadida a la tabla ARP:

```
root@Host:~# ip neighbour show 192.168.2.10
192.168.2.10 dev enp0s3 lladdr 08:00:27:ee:11:22 PERMANENT
```

El valor `PERMANENT` es el estado de la entrada y nos indica que esta entrada permanecerá en la tabla hasta que no se elimine manualmente.

### ELIMINAR ENTRADA

Con el comando `ip neighbour del` eliminamos entradas de la tabla ARP. Su sintaxis es la siguiente:

```
ip neighbour del <ip-equip> lladdr <mac> dev <interfaz>
```

Elimina manualmente una entrada de la tabla ARP indicando la IP del equipo, su MAC y la interfaz por la que se llega al equipo con esta MAC.



## Eliminar entrada de la tabla ARP

Eliminar la entrada correspondiente a la IP `192.168.2.10` de la tabla ARP que hemos añadido anteriormente.

```
root@Host:~# ip neigh del 192.168.2.10 lladdr 08:00:27:aa:aa:aa dev enp0s3
```