

## 5.1. Configuración de conexiones de red



# Índice

---

|   |    |
|---|----|
| Objetivos .....                                       | 4  |
| Configuración del protocolo TCP/IP .....              | 5  |
| Conceptos fundamentales sobre TCP/IP.....             | 5  |
| Ejemplo de formación de un paquete de datos IP .....  | 6  |
| Tipos de direcciones e identidades.....               | 6  |
| Quién las asigna y cómo son las direcciones IP .....  | 8  |
| ¿Y cómo es entonces una dirección IP? .....           | 9  |
| Clases de direcciones IP en formato IPv4 .....        | 10 |
| ¿Por qué necesitamos direcciones IPv6? .....          | 11 |
| Formato del paquete IPv6 .....                        | 13 |
| Direcciones IP y máscaras de subred .....             | 14 |
| Máscaras de subred.....                               | 14 |
| Formato de las máscaras de subred .....               | 15 |
| Direcciones de red y de “broadcast” .....             | 15 |
| Configuración estática del cliente de red IP .....    | 17 |
| IP estática en Windows.....                           | 18 |
| IP estática en Linux .....                            | 19 |
| Configuración dinámica del cliente de red IP .....    | 19 |
| Configuración de ip dinámica en windows y linux ..... | 20 |
| Importante .....                                      | 21 |
| Configuración de la resolución de nombres .....       | 21 |
| Otros sistemas: NetBIOS y WINS.....                   | 22 |
| La jerarquía de los DNS .....                         | 23 |
| Configuración de los servidores DNS .....             | 24 |
| Ficheros de configuración de red .....                | 25 |
| Tablas de enrutamiento .....                          | 27 |
| Ejemplo de tabla de enrutamiento en Linux.....        | 28 |
| Ejercicio .....                                       | 29 |
| Gestión de puertos .....                              | 30 |
| Protocolos TCP y UDP .....                            | 30 |

|  |    |
|--|----|
| Los puertos de comunicaciones .....              | 31 |
| Los puertos más conocidos .....                  | 32 |
| Gestión de los puertos en Windows y Linux .....  | 32 |
| Cómo ver los puertos abiertos en Windows .....   | 33 |
| Cómo ver los puertos y conexiones en Linux ..... | 33 |
| Ejemplo de cómo cerrar un puerto en Windows..... | 34 |
| Algunas recomendaciones .....                    | 35 |
| Despedida .....                                  | 36 |
| Resumen.....                                     | 36 |

En la actualidad prácticamente no se entiende el funcionamiento de ningún sistema sin estar conectado a la red (Internet). Por ello, es crucial para cualquier usuario o administrador del S.O. conocer cómo se realizan las conexiones y cómo se gestiona su configuración.

# Objetivos

En esta unidad perseguimos los siguientes objetivos:

1. Aprender a configurar adecuadamente los parámetros de conexión a la red como requisito indispensable para un buen funcionamiento del sistema.
2. Saber que una mala configuración puede provocar fallos y aumentar los riesgos para la seguridad del sistema.
3. Tener una visión general de los principales conceptos relacionados con la configuración de conexiones TCP/IP en los equipos, la asignación correcta de direcciones y los servidores de nombres de dominio.

# Configuración del protocolo TCP/IP

## Conceptos fundamentales sobre TCP/IP

Seguro que ya conoces la arquitectura básica TCP/IP y que existen muchos protocolos dentro de ella. Vamos a profundizar un poco en su conocimiento.

La arquitectura TCP/IP se encuentra definida en la "[RFC 1122](#)" y establece el conjunto de protocolos estructurado en cuatro capas jerarquizadas.

**Nota:** una "RFC" es una especificación oficial para la comunidad Internet que publica el "Internet Engineering Task Force" (IETF).

Resulta indispensable tener claro cómo se forman las unidades de datos (paquetes) y cómo se intercambian entre los equipos para poder configurar adecuadamente el servicio en cualquier sistema operativo. Recuerda las capas y algunos ejemplos de protocolos:

- **Nivel de aplicación:** la capa superior, que agrupa funciones de control del diálogo, codificación y representación de la información.
- **Nivel de transporte:** funciones de transporte extremo a extremo.
- **Nivel de red:** o capa de direccionamiento IP, que se encarga del enrutamiento de la información en los nodos de la red.
- **Nivel de acceso al medio:** funciones de acceso al medio, como la capa de enlace de datos y el nivel físico de OSI.

| Ej. protocolos   | MODELO TCP/IP            |
|--|--------------------------|
| HTTP, HTTPS, FTPP, Telnet, SSH, POP3, IMAP, NTP, DHCP, PING, DNS, WINS | Nivel de Aplicación      |
| TCP, UDP   | Nivel de Transporte      |
| IP, ARP, ICMP, IGMP  | Nivel de Red             |
| Ethernet, Token Ring, SDH, GSM, ATM                                    | Nivel de acceso al Medio |

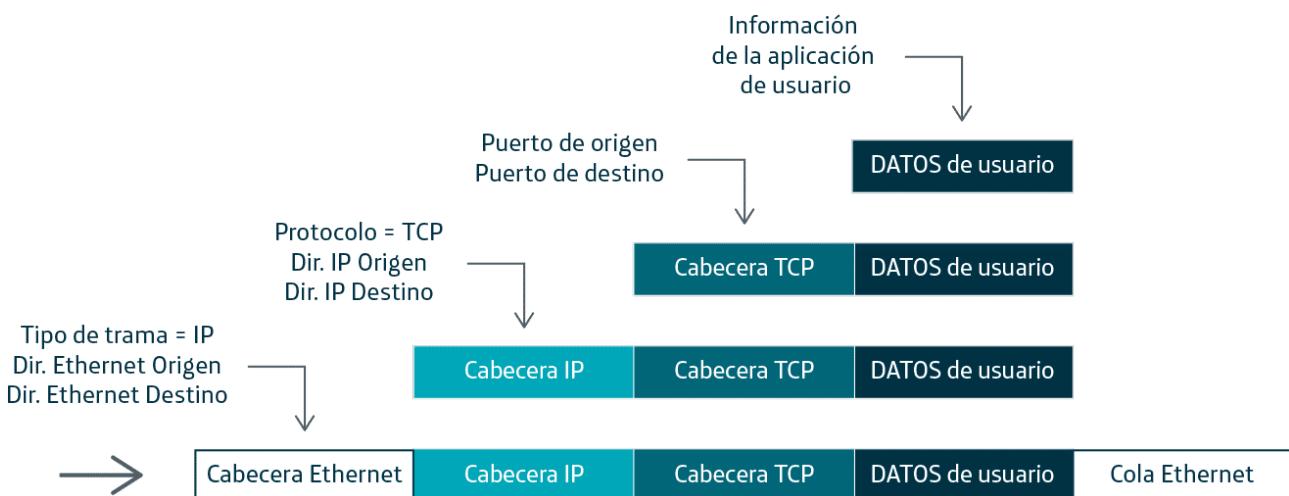
*Ejemplo de formación de un paquete de datos IP.*

## Ejemplo de formación de un paquete de datos IP

Para poder intercambiar información las aplicaciones que se encuentran corriendo en el equipo la dividirán en paquetes de datos, a los que añadirán cabeceras de protocolo, cada una de las cuales lleva parámetros que se analizarán en un determinado nivel.

En el ejemplo podemos ver cómo se construyen los paquetes con protocolo TCP/IP y se envían a través de una red local Ethernet.

Al proceso de “envolver” la información añadiéndole las cabeceras del nivel inferior se le suele llamar “encapsulado”.



## Tipos de direcciones e identidades

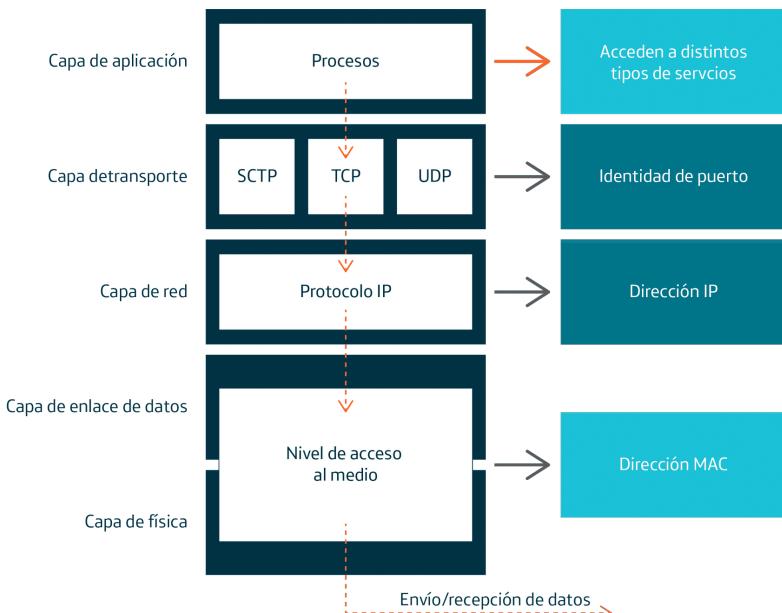
Cuando hablamos de **direcciónamiento IP** normalmente nos referimos a las direcciones manejadas en la capa/nivel de red.

Sin embargo, a la hora de gestionar nuestro sistema, debemos tener claro en cuál de los niveles estamos trabajando, y entonces podemos hablar de:

- "Punto de acceso al servicio"**

("Service Access Port") o "**Puerto**": los trabajaremos en el origen y destino de la comunicación, a nivel de transporte, y servirán de punto de acceso para los procesos que intercambian información en el "host" de origen y el "host" de destino.

- "Direccionamiento IP"**, o direccionamiento a nivel de red: cada máquina en la red tendrá su propia dirección IP (al menos una, aunque puede tener varias) que la identifica a nivel global en la red a la que está conectada (si es una "IP pública" será a nivel de red Internet).
- "Direccionamiento MAC"**: se trata del direccionamiento dentro de cada subred física a la que están conectados los equipos. Las direcciones MAC identifican físicamente una tarjeta de acceso al medio de transmisión.



*Distintos tipos de "direccionamiento".*

Por lo tanto, **todos los host (equipos) conectados a una red IP** (por ejemplo Internet) **deberán tener asignada una dirección IP** que los identifique para poder intercambiar información. Para conectar nuestro ordenador a la red tendremos que configurar los parámetros del protocolo IP en él.

El procedimiento para **configurar los protocolos TCP/IP** puede variar según el sistema operativo que estemos empleando, pero los parámetros y direcciones a emplear serán los mismos en un sistema u otro; algunos podrán auto-configurarse cuando demos de alta las conexiones de red, y otros deberemos fijarlos específicamente.



Si nuestro equipo es un simple ordenador para trabajo de oficina (un "**cliente de red**"), la configuración será más sencilla, como veremos, pero si estuviéramos trabajando con sistemas de conmutación complejos, como los *routers* del interior de la red, que necesitan gestionar sus propias tablas de encaminamiento y trabajan con protocolos de encaminamiento interiores, la gestión sería más compleja.

## Quién las asigna y cómo son las direcciones IP

Debemos distinguir entre una red privada (que se gestiona de forma particular en nuestro entorno) y una red pública (conectada a Internet).

Si la red es privada y está aislada el propio administrador podrá fijar las direcciones a configurar en los equipos. Pero en el momento en que queramos conectarnos a la red pública, es decir, Internet, deberemos compartir el esquema de direccionamiento con los demás accesos a nivel mundial.

Es por esto que para asignar las direcciones IP de Internet la autoridad máxima es la **IANA** ("Internet Assigned Number Authority"), que a su vez es un departamento dependiente de la **ICANN** ("Internet Corporation for Assigned Names and Numbers"). A partir de ellas, se delega la gestión de grandes bloques de direcciones IP a los denominados registros regionales, de los que, de momento, existen tres en el mundo: **RIPE**, **ARIN**, y **AP-NIC**.

**RIPE NCC** ("RIPE Network Coordination Center") es el registro delegado de Internet a nivel europeo y se encarga, entre otras tareas, de la asignación de bloques de direcciones IP a los proveedores de servicios de Internet en Europa y su área de influencia.



Después, las organizaciones, empresas y usuarios finales reciben las direcciones IP necesarias para conectarse a Internet a través de su **proveedor de acceso a Internet**, quien a su vez las habrá obtenido, bien de su proveedor de tránsito, bien del registro regional correspondiente.

Cuando nuestra organización contrata el servicio de acceso, o el hospedaje de páginas en un servidor, también podrá contratar una serie de "IP públicas" para utilizar como origen/destino a nivel global.

## ¿Y cómo es entonces una dirección IP?

A nivel de red el encaminamiento de los datos por la red utiliza el protocolo IP, de forma que **cada equipo conectado a la red (a los que llamamos comúnmente "hosts") tendrá al menos una dirección IP que lo identificará de forma única** entre todos los conectados a la misma red.

Aunque más adelante lo veremos con más detalle, podemos ver que las direcciones **IP (IPv4)** están formadas por una secuencia de **32 bits**, es decir, cuatro octetos que se suelen expresar en notación decimal separados por puntos. Por ejemplo, la dirección IP pública de la “Asociación de Usuarios de Internet” (esto es, la del servidor web que almacena su página y nos la devuelve cuando se la solicitamos) es la siguiente:

### Dirección IP

**82.223.248.111**  
**01010010.11011111.11111000.01101111**

Evidentemente, esta dirección puede ser permanente o en algunos casos puede cambiar con el tiempo, como cuando se reubica un *host* que hace de servidor web, en cuyo caso deberán actualizarse las tablas de los **servidores de traducción de nombres a IP** (los que llamamos **servidores de nombre de dominio** o “DNS – Domain Name Server”).

**Nota: la separación con puntos es un convenio para representarla; dentro de la máquina una dirección solo es un código binario.**

Dentro de todo el conjunto de redes IP, a la hora de direccionarlas podemos agruparlas en función de sus direcciones. Entonces hablamos de que existen diferentes **“subredes IP”** interconectadas.

Este formato de direcciones IP que hemos visto, de 32 bits, corresponde a la **versión IPv4 del protocolo**. Dentro de los bits que forman la dirección IPv4, unos cuantos identificarán a la **“subred”** en la que está el equipo y otros identificarán al propio equipo (**“host”**) dentro de esa red (subred).

El número de bits disponible para identificar el *host* dependerá del número de equipos que tengamos dentro de una “subred”. Esto da lugar a que tengamos varios “tipos” (**clases**) de direcciones IP en función de lo grandes que sean las subredes (es decir, su nivel dentro de la jerarquía global), y por tanto varios tipos de “subredes”.



*Estructura general de una dirección IP.*

A estos diferentes tipos de direcciones se les llama “**clases**” de direcciones IP. En IPv4 tenemos cinco: A, B, C, D y E, que se corresponden con sus equivalentes “tipos de redes” y que veremos a continuación.

## Clases de direcciones IP en formato IPv4

Existen diferentes clases de direcciones IP. En IPv4 tenemos cinco: A, B, C, D y E, que se corresponden con los equivalentes tipos de redes.

Comenzaremos analizando el formato de direcciones IPv4.

Como vemos, las direcciones IP, en su formato, tienen dos partes: una **identidad de red** y la **identidad del nodo** dentro de esa red. En función del “tamaño” (cantidad de nodos) de la subred en cuestión puede haber varios tipos de direcciones: “**clase A**”, “**clase B**”, “**clase C**”, “**clase D**”, “**clase E**”. Para representarlas gráficamente serían:

|         |                            |                 |                                   |
|---------|----------------------------|-----------------|-----------------------------------|
| CLASE A | Número de red<br>0 -127    | Dirección local | 16.777.216<br>direcciones locales |
| CLASE B | Número de red<br>128 - 191 | Dirección local | 65.536<br>direcciones locales     |
| CLASE C | Número de red<br>192 -223  | Dirección local | 256<br>direcciones locales        |
| CLASE D | Número de red<br>224 -239  |                 | Direcciones de multienvío         |
| CLASE E | Número de red<br>240 -255  |                 | Uso reservado                     |

*Representación gráfica de las cinco clases de direcciones IP.*

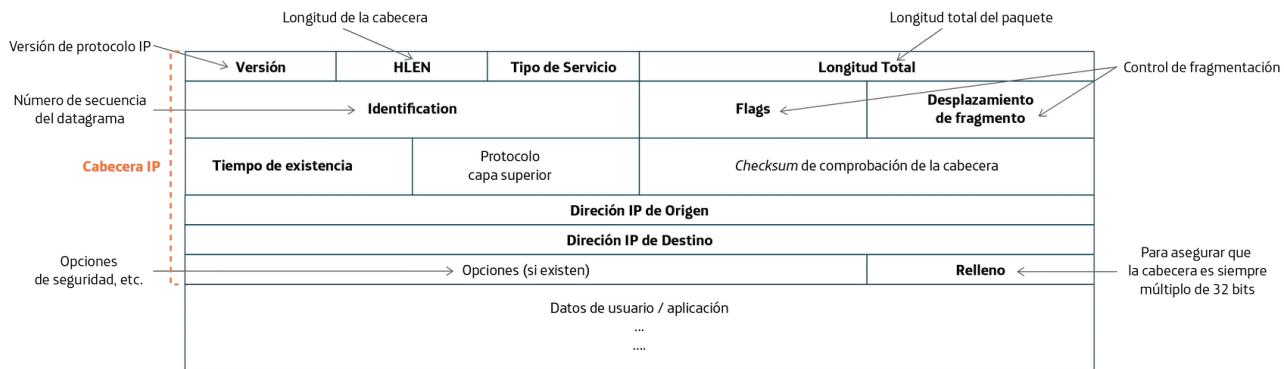
## Direcciones especiales

Dentro del rango de posibles numeraciones disponibles para una subred, no todas se pueden asignar a los equipos (host) conectados a ella. Ciertas direcciones se reservan para usos especiales. Por ejemplo:

- Si todos los bits destinados a identificar al "host" se ponen a "1" tenemos la **dirección de difusión ("broadcast")**: lo enviado a ella llega a todos los equipos de esa subred. Por ejemplo en una de clase C: 110x.xxxx.xxxx.1111 = X.X.X.255 (en decimal)
- La dirección **127.0.0.1** se denomina "dirección de bucle de retorno" e identifica al propio equipo (*host local*).
- La red 0.0.0.0 no existe.
- Cuando los bits destinados a identificar a los equipos se dejan todos a cero, el número resultante es el "**identificador de red**". Por ejemplo 126.0.0.0 identificaría a toda una red de Clase A, y 128.0.0.0 a toda una red de Clase B.

## ¿Y cómo es el paquete de datos IPv4?

La estructura básica de un paquete IPv4 te la mostramos en la siguiente figura:



## ¿Por qué necesitamos direcciones IPv6?

El protocolo IPv4 se ha quedado pequeño para la gran evolución de Internet. Ya a principios de 2010 quedaban menos del 10% de direcciones IPv4 (de 32 bits) sin asignar. En febrero de 2011 la IANA entregó el último bloque de direcciones disponibles (33 millones) en ese momento.

**IPv6**, definido en el "**RFC 2460**", es la versión moderna del protocolo IP, y está destinada a sustituir la versión IPv4, que todavía se emplea mayoritariamente.

El nuevo protocolo permite extender el rango de direcciones y asignar una identificación IPv6 prácticamente a cualquier dispositivo (¿has oído hablar del "Internet de las cosas"? "**Internet of Things**" - IOT). Tengamos en cuenta que mientras que IPv4 puede direccionar hasta  $2^{32}$  dispositivos, IPv6 permite direccionar hasta  $2^{128}$ , es decir... ¡340 sextillones de direcciones!

Las direcciones **IPv6**, de **128 bits** de longitud, se escriben como ocho grupos de cuatro dígitos hexadecimales, por ejemplo:

**"200A:0CD8:85B3:08F3:0000:8A2E:0370:7334"**

Que equivale a una dirección en código binario:

0010 0000 0000 1010 0000 1100 1101 1000 1000 0101 1011 0011 0000 1000 1111  
0011 0000 0000 0000 1000 1010 0010 1110 0000 0011 0111 0000 0111 0011  
0011 0100

Aunque si alguno de los grupos de palabras es nulo (=cero) pueden usarse notaciones abreviadas utilizando ":" o poniendo solo un "cero" en el grupo en cuestión. Puedes encontrar varios ejemplos con detalles en: <http://es.wikipedia.org/wiki/IPv6>.

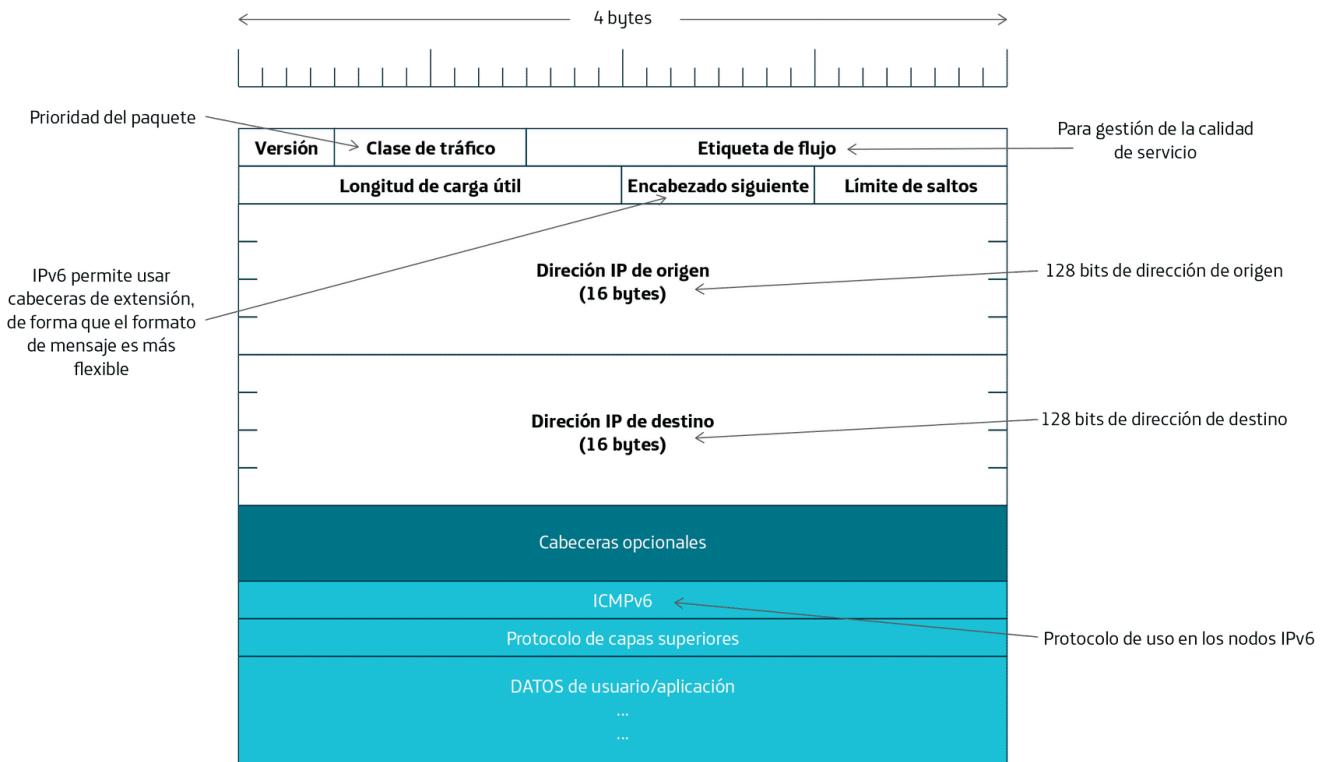
Además de extender el rango de direccionamiento IPv6, la cabecera del mensaje se simplificó y optimizó en sus campos, de forma que ofrece mayores ventajas a nivel de organización de subredes y encaminamiento en los **router**, como por ejemplo:

- **Optimización del encaminamiento** en la red troncal.
- **Optimización del direccionamiento *multicast*** y posibilidad de direccionamiento ***anycast***.
- Capacidades de autoconfiguración en los nodos.
- **Mayor seguridad** intrínseca en el núcleo del protocolo.
- Implementación de **niveles de calidad de servicio y clases de servicios**.
- **Flexibilidad** en los paquetes de datos (eficientes y extensibles).
- **Renumeración** (de direcciones locales a globales) y ***multihoming*** (conexión a más de una red).
- Apropiado para los **dispositivos móviles**.



# Formato del paquete IPv6

Como ves en la figura, se ha aumentado mucho el campo de direccionamiento y el formato es mucho más completo:



*Paquete de datos IPv6.*

Se recomienda ver <https://www.rfc-es.org/rfc/rfc2460-es.txt>.

# Direcciones IP y máscaras de subred

## Máscaras de subred

Como sabemos, si nuestro equipo está conectado a una red IP debe estar identificado por una dirección IP. Existen varias clases de subredes en función del rango de direcciones que puedan tener.

La **máscara de subred** es una combinación de bits (con el mismo formato que las direcciones) que nos sirve para delimitar el rango de direcciones de la red, es decir, su tamaño (entendido como el número posible de direcciones IP dentro de la subred).

La **máscara de subred** solamente contendrá “unos” y “ceros” consecutivos, de forma que a partir de ella la primera dirección se corresponderá con la **“dirección de red”** y la última del rango con la **“dirección de difusión (“broadcast”)** dentro de la subred.

La notación de la máscara de subred a menudo se expresa en decimal. Por ejemplo, si la máscara de subred fuese:

**11111111.11111111.11111111.11000000**

Sería igual a:

**255.255.255.192**

Por otro lado, como la máscara está directamente relacionada con el número máximo de direcciones IP que puede haber en la subred, si nos dijeran, por ejemplo, que nuestra red solo puede tener 64 direcciones IP, entonces ya conoceríamos cuál debe ser la máscara de subred, porque sería aquella que nos dejase 64 posibles direcciones libres en su rango para los equipos a conectar (en este caso es la del ejemplo que acabamos de poner).

## Formato de las máscaras de subred

Podemos entonces visualizar en el siguiente cuadro los diferentes tipos de máscaras de subred y cuál sería el máximo de direcciones IP que puede haber en esas subredes:

| Máscaras de subred                  |                  | Nº dir IP | Número máximo de direcciones IP en esa subred.<br>Recordemos que la primera y la última están reservadas. |
|-------------------------------------|------------------|-----------|---|
| Notación en binario                 | Notación decimal |           |   |
| 11111111.00000000.00000000.00000000 | 255.0.0.0        | 16777216  |   |
| 11111111.10000000.00000000.00000000 | 255.128.0.0      | 8388608   |   |
| 11111111.11000000.00000000.00000000 | 255.192.0.0      | 4194304   |   |
| 11111111.11100000.00000000.00000000 | 255.224.0.0      | 2097152   |   |
| 11111111.11110000.00000000.00000000 | 255.240.0.0      | 1048576   |   |
| 11111111.11111000.00000000.00000000 | 255.248.0.0      | 524288    |   |
| 11111111.11111100.00000000.00000000 | 255.252.0.0      | 262144    |   |
| 11111111.11111110.00000000.00000000 | 255.254.0.0      | 131072    |   |
| 11111111.11111111.00000000.00000000 | 255.255.0.0      | 65536     |   |
| 11111111.11111111.10000000.00000000 | 255.255.128.0    | 32768     |   |
| 11111111.11111111.11000000.00000000 | 255.255.192.0    | 16384     |   |
| 11111111.11111111.11100000.00000000 | 255.255.224.0    | 8192      |   |
| 11111111.11111111.11110000.00000000 | 255.255.240.0    | 4096      |   |
| 11111111.11111111.11111000.00000000 | 255.255.248.0    | 2048      |   |
| 11111111.11111111.11111100.00000000 | 255.255.252.0    | 1024      |   |
| 11111111.11111111.11111110.00000000 | 255.255.254.0    | 512       |   |
| 11111111.11111111.11111111.00000000 | 255.255.255.0    | 256       |   |
| 11111111.11111111.11111111.10000000 | 255.255.255.128  | 128       |   |
| 11111111.11111111.11111111.11000000 | 255.255.255.192  | 64        |   |
| 11111111.11111111.11111111.11100000 | 255.255.255.224  | 32        |   |
| 11111111.11111111.11111111.11110000 | 255.255.255.240  | 16        |   |
| 11111111.11111111.11111111.11111000 | 255.255.255.248  | 8         |   |
| 11111111.11111111.11111111.11111100 | 255.255.255.252  | 4         |   |

Si deseas ampliar información puedes visitar el siguiente [enlace](#).

## Direcciones de red y de “broadcast”

Si nos dan una dirección IP y conocemos la máscara de subred, resultará fácil averiguar las **direcciones de red y de difusión** para esa red. Recordemos que la **dirección de red** es la primera (todo ceros en los bits de dirección utilizables en esa red) y la de difusión (“broadcast”) la última (todo unos en los bits de dirección utilizables en esa red).

**Veámoslo con un ejemplo:**

Si sabemos que la IP de nuestro equipo es “192.168.2.101” y la máscara de subred es “255.255.255.192”, entonces:

- Para calcular la dirección de red hacemos un “AND” lógico entre la dirección IP y la máscara:**

```
Dir IP: 11000000 10101000 00000010 01100101
Máscara: 11111111 11111111 11111111 11000000
-----
AND lógico: 11000000 10101000 00000010 01000000
Dir. De red 192.168.2.64
```

- Para calcular la dirección de difusión hacemos un “OR” lógico entre la dirección IP y el inverso máscara (negación):**

```
Dir IP: 11000000 10101000 00000010 01100101
Máscara: 00000000 00000000 00000000 00111111
-----
OR lógico: 11000000 10101000 00000010 01111111
Dir. De red 192.168.2.127
```

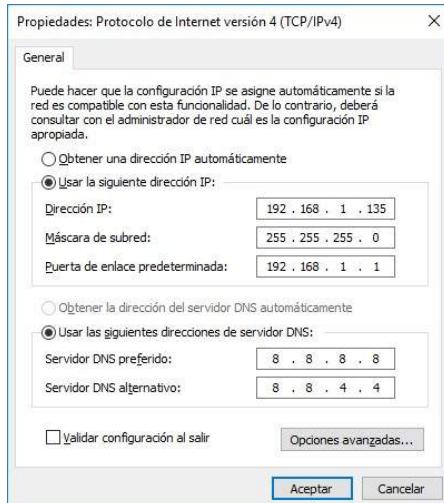
**¿Y cómo funciona en IPv6?**

En IPv6 podemos manejar también el concepto de máscara, pero teniendo en cuenta que **el tamaño de una subred en IPv6 es de 264 direcciones**, es decir, una **máscara de subred de 64-bit**, lo cual permitirá construir subredes mucho más grandes y administrar el enrutamiento de forma más eficiente, con rutas de mayor capacidad de direccionamiento, etc.

# Configuración estática del cliente de red IP

Cuando asignamos de forma fija una dirección IP a nuestro *host* decimos que tiene una "configuración estática", es decir, que no cambia de una sesión de trabajo a otra.

De este modo, nuestro equipo tendría una "**IP fija**" y siempre usaría esa dirección para conectarse a la red.



La configuración estática también incluye, evidentemente, la máscara de subred y el "**gateway**" (puerta de enlace) por defecto.

También podemos definir que se utilicen siempre unos **DNS** ("Domain Name Server" = traductor de nombres a direcciones IP) o que se obtengan automáticamente (si disponemos de un servidor en la red que nos proporciona el servicio).

*Configuración IP estática.*

## VENTAJAS

Disponer de una IP fija puede ser necesario en algunos casos. Por ejemplo si ciertas aplicaciones necesitan tener asociada de forma permanente una dirección IP y un puerto, o si deseamos dar a conocer esa IP al resto de la red para acceso sin necesidad de tener asociado un nombre de dominio.

También **podemos ganar en seguridad**, pues en el momento en que otro equipo quisiera conectarse a la red con esa dirección nos alertaría con un conflicto de direcciones IP, y la gestión de la red puede ser más sencilla, o al menos aportar un **mayor control** al tener **identificado un equipo concreto por su IP asignada**.

## INCONVENIENTES

Sin embargo, en la **configuración estática** de la IP tenemos el inconveniente de necesitar disponer de tantas IP como equipos tengamos, estén o no en funcionamiento, y esto no es sencillo si además las direcciones IP han de ser públicas (pues como sabemos estas nos las tienen que asignar, al estar controladas a nivel global en la red).

Por otra parte, la planificación de la red ha de hacerse cuidadosamente, pues **dos equipos no pueden tener la misma dirección IP** y estar en funcionamiento al mismo tiempo (lo cual se complica si algún equipo tiene más de una dirección asignada), ni debe permitirse que el usuario pueda cambiar su dirección por su cuenta.

La configuración estática también incluye evidentemente la máscara de subred y el “*gateway*” (puerta de enlace) por defecto.

**Ejercicio:** en la figura hemos puesto como DNS dos muy particulares. ¿Te atreves a averiguar de quién son?, es decir, ¿a quién estamos configurando para que traduzca URL a direcciones IP? Simplemente búscalos en Internet :-).

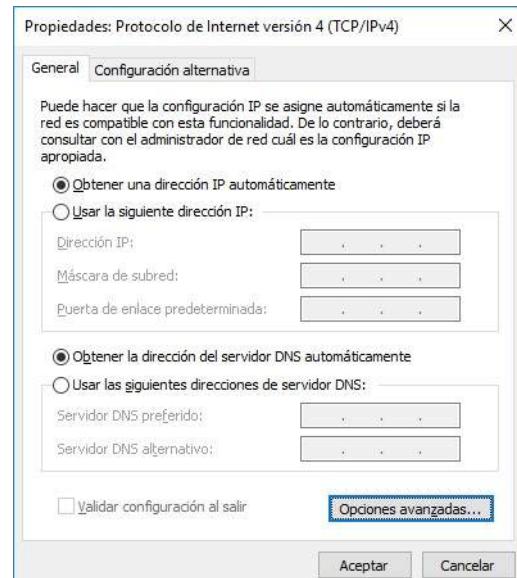
## IP estática en Windows

En Windows la configuración de la dirección IP se realiza a través de las propiedades del protocolo en cada una de las conexiones. Esto es, con algunas pequeñas diferencias entre versiones, si estamos conectados vía interfaz de red Ethernet, un ejemplo del camino a seguir sería:

*Botón inicio > Panel de control > Redes e Internet > Centro de redes y recursos compartidos > Conexión de área local > Propiedades > Elegimos el protocolo IPv4 o IPv6 > Propiedades*

Por ejemplo, en **Windows 10** nos saldría una ventana como la de la imagen de al lado, en la que también podemos configurar la máscara de subred, la puerta de enlace predeterminada y los DNS, como acabas de ver un poco más arriba en esta misma lección.

Como sabemos, si asignamos IP fijas a los equipos de nuestra red, deberemos asegurarnos de que elegimos direcciones diferentes para todos ellos y no repetimos ninguna, lo cual provocaría problemas en la red cuando ambos equipos estuviesen conectados al mismo tiempo.



Si queremos visualizar la configuración IP que tiene nuestro equipo desde la consola de comandos podemos introducir el comando “**ipconfig /all**”, y nos mostrará los datos de la conexión.

## IP estática en Linux

Como siempre, debemos tener en cuenta que podemos encontrar algunas diferencias entre las "distros" de Linux. En general se pueden configurar las conexiones de red a través de comandos y ficheros de configuración o también a través del entorno gráfico.

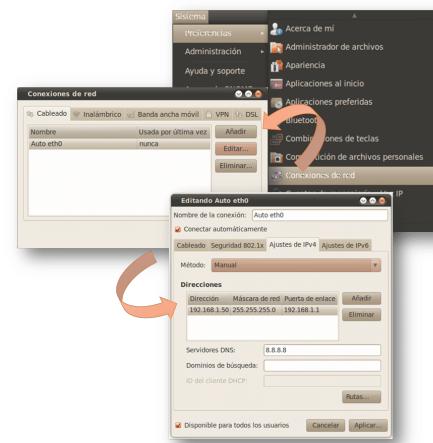
En el caso de Ubuntu, si elegimos hacerlo usando el entorno gráfico podemos hacerlo a través de:

*Sistema > Preferencias > Conexiones de red > elegimos el tipo de conexión (p. ej. "cableado" para la red local Ethernet) > Editar...*

Y nos saldrá una ventana con las opciones para insertar los datos de la dirección IP, máscara de subred, puerta de enlace y DNS.

A través de un terminal podemos visualizar la configuración con el comando "ifconfig", y para modificar la IP y el resto de parámetros podemos editar el fichero de configuración: "/etc/network/interfaces", en el que podremos configurar una IP fija para cada una de las conexiones del equipo (red local, inalámbricas, etc.).

Después de hacerlo, deberemos reiniciar las interfaces de red (por ejemplo con el comando: "/etc/init.d/networking restart").

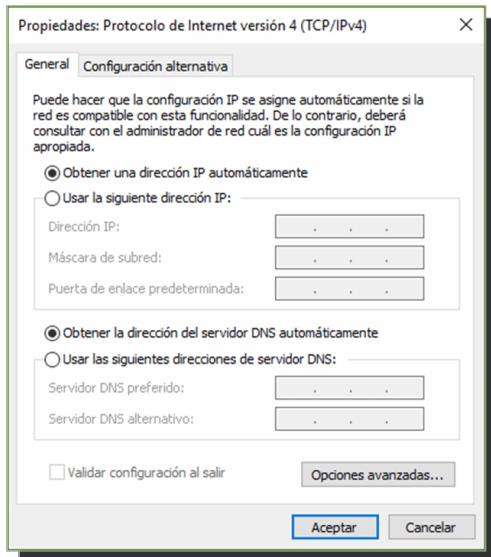


## Configuración dinámica del cliente de red IP

Cuando se asigna una configuración dinámica automática en las direcciones IP, normalmente se hace a través del protocolo DHCP ("Dynamic Host Configuration Protocol").

En este caso, en la red debe existir un servidor DHCP, que se encargará de suministrar los parámetros de conexión a los equipos que se lo soliciten, y entre ellos su dirección IP.

Esta configuración (la asignación de una dirección IP) se realiza en el momento en que el equipo se conecta a la red. A menudo, en las instalaciones domésticas, el servidor DHCP es el propio router de conexión a Internet.



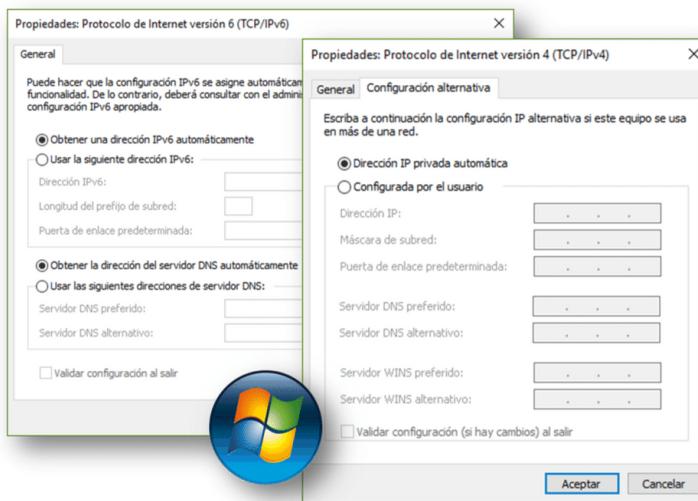
Si se usa la asignación dinámica se puede reducir el número de direcciones disponibles, ya que solamente necesitarán tener IP asignada aquellos equipos conectados a la red y que estén en funcionamiento. También es posible disponer de tablas de asignación prefijadas en el servidor DHCP para que asigne determinadas direcciones a los equipos en función de su dirección física (dirección MAC Ethernet).

Lo más común es que el **propio equipo del usuario** **solicite una dirección** en el momento de conectarse, y el DHCP se la asigne dinámicamente para esa sesión de trabajo.

Si deseas ampliar información puedes consultar más información sobre el protocolo DHCP - RF2131 en el siguiente [enlace](#).

## Configuración de ip dinámica en windows y linux

La configuración de una dirección IP dinámica, tanto en Windows como en Linux, se realiza a través de las mismas opciones que hemos visto para configurar la dirección IP estática, pero ahora en la opción de configuración elegiremos “obtener la dirección automáticamente”, por ejemplo.



*Ejemplo de configuración de una dirección IP dinámica en Windows y en Ubuntu.*

## Importante

El hecho de que una **IP sea pública o privada** depende de si pertenece al esquema de direccionamiento global de la red (entonces es pública) o bien a un esquema interno de numeración dentro de la organización (entonces es privada).

Por otra parte, que sea “**IP fija**” significa que el equipo la tiene asignada de forma permanente y no cambia de una sesión de trabajo a otra, mientras que si es una “**IP dinámica**” significa que se le asigna una distinta cada vez que se conecta o solicita una dirección IP (o también si se reinicia el servidor DHCP por ejemplo).

**¡Ojo! No debemos confundir “IP pública” con “IP fija”, o “IP privada” con “IP dinámica”.**

## Configuración de la resolución de nombres

El sistema de resolución de nombres de dominio se encarga de averiguar la dirección IP asociada a un nombre de dominio (“dirección web”), de forma que podamos navegar por Internet utilizando estos nombres en lugar de recordar direcciones numéricas.

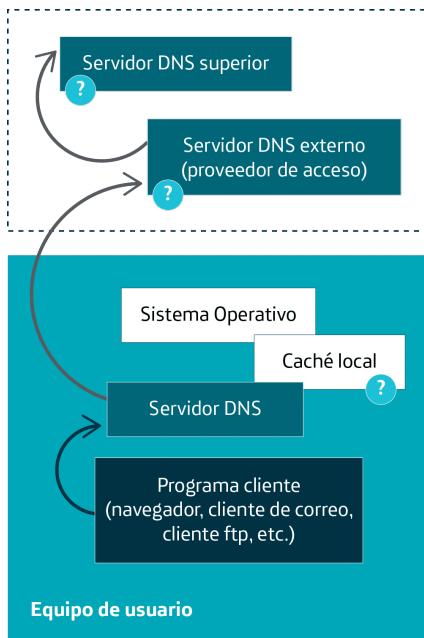
Los servidores de nombres DNS proporcionan el servicio de conversión de nombre a dirección IP.

Las direcciones IP son asignadas a equipos conectados a la red que están identificados también por un nombre (“**hostname**”), pero normalmente no conocemos su dirección IP y lo que hacemos es pedir la conexión (por ejemplo en el navegador) con una dirección web.

La resolución de nombres se realiza de forma transparente para el usuario, pues es solicitada por la aplicación cliente que esté manejando (el navegador, cliente de correo, etc.).

### Pasos a seguir en la resolución de nombres

1. Cuando la aplicación cliente necesita traducir la dirección, primero comprueba si se encuentra en la memoria caché local.
2. Si no lo encuentra, solicita la traducción al S.O., que comprueba si existe un servicio de DNS local.
3. Si no lo encuentra consulta sus ficheros de configuración para hacer la petición a un DNS externo.
4. El servidor DNS al que se ha hecho la solicitud busca la traducción, y si no la encuentra elevará la petición a un servidor de orden superior.



Es decir, la estructura de DNS sigue una **organización jerárquica y en forma de “árbol”**, donde los servidores tienen cada uno un tipo de “dominio” o parte de él. Desde los **servidores raíz** de nivel superior se van agrupando otros dependientes de ellos y estos a su vez pueden tener a otros por debajo.

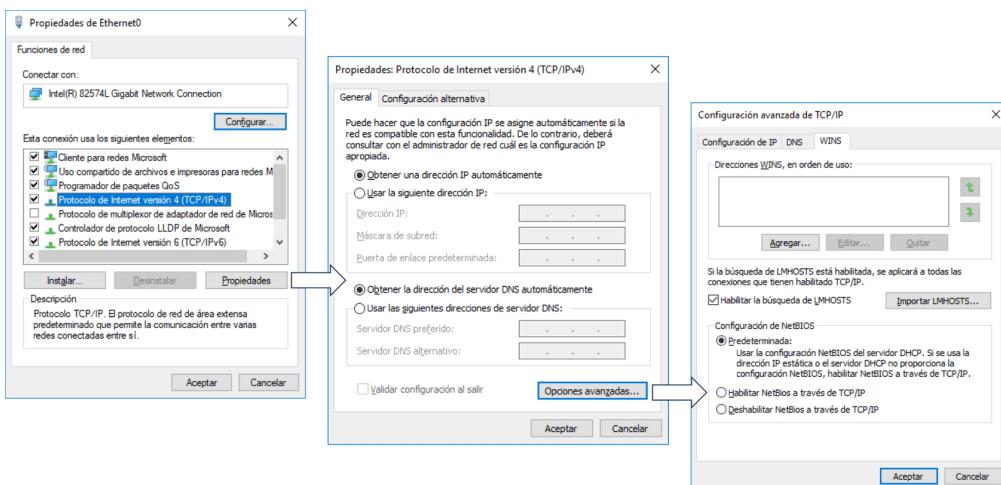
Si un DNS no encuentra la correspondencia y no puede realizar la traducción pasará la “pregunta” a otro y este a su vez hará lo mismo hasta obtener la dirección IP asociada al nombre de dominio.

## Otros sistemas: NetBIOS y WINS

Aunque el sistema de nombres de dominio (DNS) es el utilizado ampliamente hoy en Internet, no ha sido el único. Existen otros sistemas que permitían asociar un nombre a un equipo.

Por ejemplo, antiguamente los sistemas Windows utilizaban el protocolo **“NetBIOS”**, que asociaba un nombre lógico de 16 caracteres a la dirección física de cada tarjeta de red, almacenándolos en una tabla de nombres. Los nombres NetBIOS podían ser individuales o de grupo, y debían ser únicos en la red.

Cuando un “host” se inicializa envía un mensaje de difusión solicitando ser registrado con un nombre, y si no existe otro igual recibirá una confirmación, mientras que si hubiera otro equipo ya con ese nombre reportaría un fallo en la inicialización.



El problema que presentaba NetBIOS era precisamente que se basaba en el registro mediante el envío de mensajes de difusión que no soportaban la existencia de rutas y encaminamiento a través de routers.

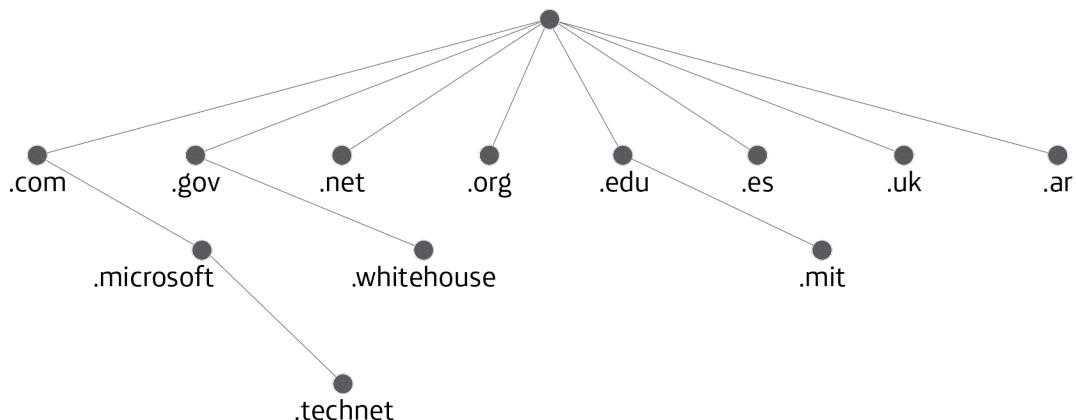
Para solucionar esto Microsoft creó un nuevo protocolo, el **WINS** ("Windows Internet Naming Service"), que proporcionaba una base de datos distribuida que registraba las asignaciones dinámicas de nombres NetBIOS y los hacía corresponder a direcciones IP. Los nombres están registrados en un **servidor WINS** que responde a la resolución de nombres con direcciones IP registradas en su base de datos. En la red podrán existir varios servidores WINS para trabajar en reparto de carga, que periódicamente actualizan sus tablas para que los datos sean coherentes en todos ellos.

## La jerarquía de los DNS

La jerarquía de los DNS parte del **nivel raíz**, en el cual están los DNS dedicados a los dominios territoriales (por ejemplo ".es", ".uk", "mx") y los dominios genéricos de nivel superior para los diferentes tipos de organizaciones ("com", ".edu", ".org").

A partir de ellos dependen otros para un grupo determinado de subdominios: por ejemplo Microsoft es la autoridad designada por los servidores raíz de Internet para su propia parte del árbol del espacio de nombres de dominio DNS en Internet.

Un ejemplo de estructura podría ser:



Estructura en forma de árbol de una DNS.

En la figura anterior, por ejemplo, habría un DNS para todos los dominios ".com", del cual depende otro encargado de todos los dominios ".microsoft" y a su vez de este depende uno para todos los dominios ".technet".

Es decir, la estructura de DNS sigue una **organización jerárquica** y en forma de árbol, donde los servidores tienen cada uno un tipo de “dominio” o parte de él. Desde los servidores raíz de nivel superior se van agrupando otros dependientes de ellos y estos a su vez pueden tener a otros por debajo.

Puedes ampliar la información en los siguientes enlaces:

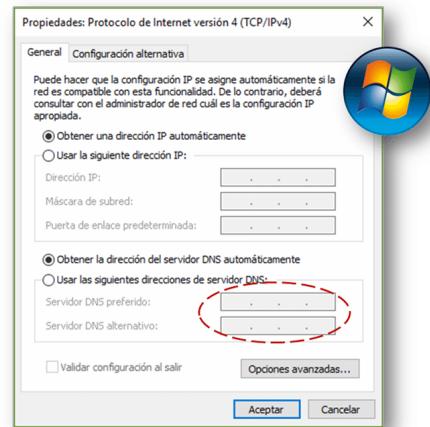
[https://technet.microsoft.com/es-es/library/dd197427\(v=ws.10\).aspx](https://technet.microsoft.com/es-es/library/dd197427(v=ws.10).aspx)

[https://es.wikipedia.org/wiki/Domain\\_Name\\_System](https://es.wikipedia.org/wiki/Domain_Name_System)

## Configuración de los servidores DNS

### Entorno Windows

Para realizar la configuración de los servidores DNS en nuestro equipo, **si estamos trabajando en entorno Windows lo haremos a través de la misma opción que para configurar la dirección IP**, y de nuevo podremos elegir entre la configuración automática vía DHCP o introducir unas direcciones IP de los DNS preferidos para solicitar las traducciones.



Configuración de los servidores DNS en Windows.



### Entorno Linux - Ubuntu

En el caso de Linux, podemos hacerlo también a través del entorno gráfico, o bien a través de un terminal de consola, editando el archivo “*/etc/resolv.conf*”, por ejemplo poniendo:

**# Configuración de servidores DNS  
nameserver 8.8.8.8**

Y reiniciando las conexiones de red:

**\$ sudo /etc/init.d/networking restart**

Configuración de los servidores DNS en Linux.

# Ficheros de configuración de red

En muchos sistemas operativos es posible realizar la configuración de los parámetros de red editando directamente ciertos ficheros.

Pero de nuevo encontramos diferencias entre las distintas familias de sistemas operativos y entre las diferentes generaciones dentro de cada familia.

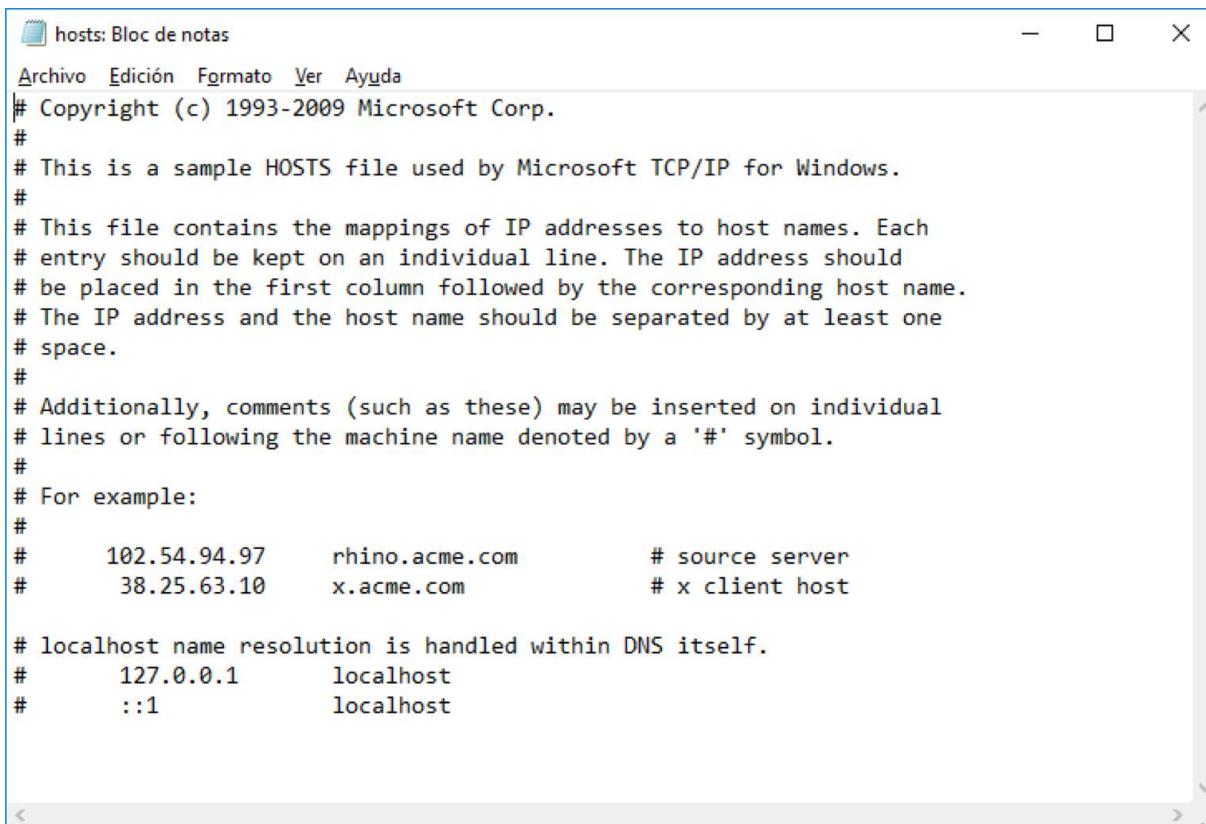
Veamos cómo funciona la configuración de parámetros de red en algunos sistemas operativos, como Windows y Linux Ubuntu.

## WINDOWS

En el caso de Windows la configuración se suele realizar a través de las opciones del entorno gráfico, sin embargo existen algunos archivos en los que podemos guardar datos de configuración.

Por ejemplo en los archivos “**Hosts**” o “**LMHosts**”; el primero para utilidades de TCP/IP y el segundo para utilidades de LAN Manager, y en los que se puede guardar una primera tabla de traducciones de nombres a direcciones IP.

En la imagen vemos el Archivo “Hosts” ubicado en: *C:\Windows\System32\drivers\etc*



The screenshot shows a Windows Notepad window titled "hosts: Bloc de notas". The window contains the following text:

```
hosts: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com      # source server
#      38.25.63.10      x.acme.com          # x client host
#
# localhost name resolution is handled within DNS itself.
#      127.0.0.1        localhost
#      ::1              localhost
```

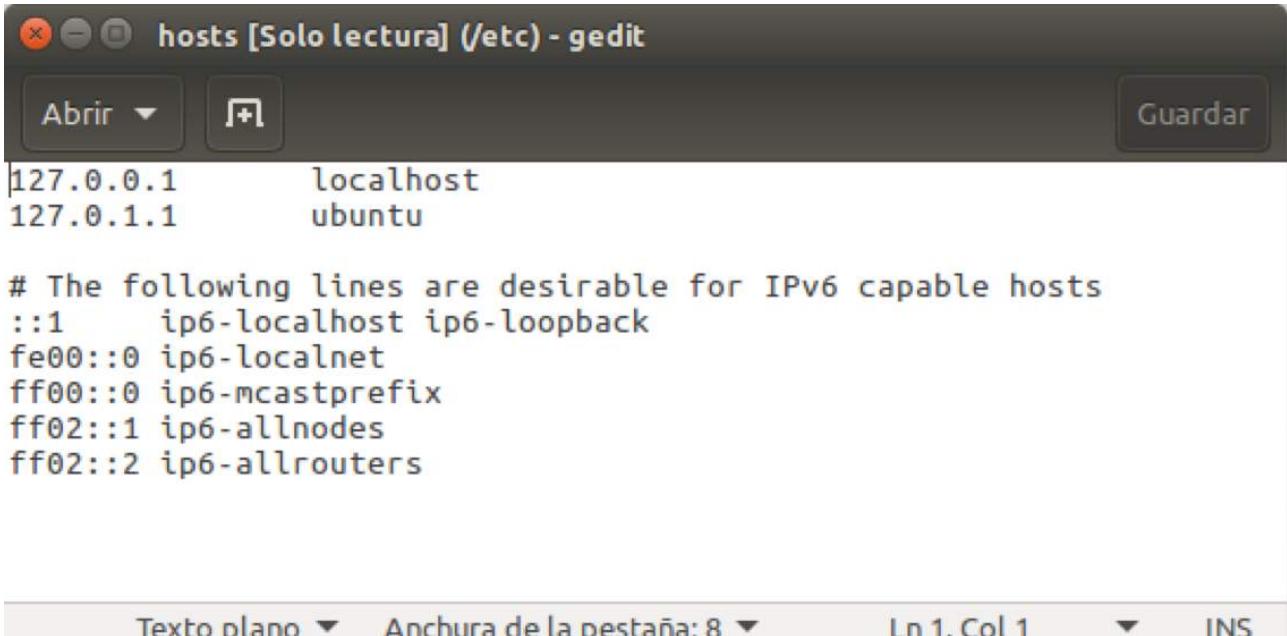
## EJEMPLO LINUX UBUNTU

En Linux Ubuntu la configuración de red se guarda en varios ficheros, como por ejemplo:

“**/etc/network/interfaces**” o “**/etc/hosts**”. En ellos se guardarán, por ejemplo:

- El tipo de inicio (automático, manual o asistido).
- El tipo de configuración (IP estática, DHCP, etc.).
- La dirección IP de red y la máscara de red asociada (salvo si se configura automáticamente por DHCP).
- Los ficheros “scripts” a ejecutar en el inicio (*ifconfig interface up*) y en la parada (*ifconfig interface down*).
- Equivalencia local entre nombres y direcciones IP.

En la imagen vemos un ejemplo del fichero **/etc/hosts**.



The screenshot shows the gedit text editor window titled "hosts [Solo lectura] (/etc) - gedit". The window contains the following text:

```
127.0.0.1      localhost
127.0.1.1      ubuntu

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

At the bottom of the window, there are status indicators: "Texto plano ▾" (Plain text), "Anchura de la pestaña: 8 ▾" (Tab width: 8), "Ln 1, Col 1 ▾" (Line 1, Column 1), and "INS" (Insert mode).

## Tablas de enrutamiento

Una tabla de enrutamiento (o encaminamiento) es un conjunto de datos que identifica las posibles rutas para alcanzar un destino ("host" o nodo) dentro de la red.

Las tablas de enrutamiento se almacenarán normalmente en aquellos puntos que tienen que encaminar la información (p. ej. un *router*), eligiendo la mejor ruta de entre las disponibles.

Si la red es pequeña y no sufre muchos cambios, las tablas de enrutamiento pueden mantenerse manualmente por el administrador. Sin embargo, en redes grandes se suele disponer de **capacidad de enrutamiento dinámico, y los nodos configuran y actualizan de forma automática sus propias tablas de enrutamiento**. Para lograrlo se intercambia información entre los propios nodos de conmutación, relativa a la topología y el estado de la red, los fallos existentes y las rutas disponibles, o bien las rutas en situación de congestión si eso se produce. Es decir, los nodos (*router*) de la red son capaces de adaptarse automáticamente a los cambios dentro de la propia red.

Por ejemplo, para ver la tabla de enrutamiento de nuestro router podemos acceder al perfil de administración que posee el *router*, y también podemos ver las que tiene nuestro sistema mediante comandos, como por ejemplo "**route print**" (o "netstat -r") desde la consola de Windows.

```
C:\> Símbolo del sistema
C:\Users\Alumno>route print
=====
ILista de interfaces
 4...00 0c 29 59 fc 60 .....Intel(R) 82574L Gigabit Network Connection
 1........................Software Loopback Interface 1
 2...00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
=====
IPv4 Tabla de enrutamiento
=====
Rutas activas:
Destino de red   Máscara de red   Puerta de enlace   Interfaz   Métrica
 0.0.0.0         0.0.0.0        192.168.186.2    192.168.186.131  25
 127.0.0.0       255.0.0.0      En vínculo        127.0.0.1     331
 127.0.0.1       255.255.255.255 En vínculo        127.0.0.1     331
 127.255.255.255 255.255.255.255 En vínculo        127.0.0.1     331
 192.168.186.0   255.255.255.0 En vínculo        192.168.186.131  281
 192.168.186.131 255.255.255.255 En vínculo        192.168.186.131  281
 192.168.186.255 255.255.255.255 En vínculo        192.168.186.131  281
 224.0.0.0        240.0.0.0      En vínculo        127.0.0.1     331
 224.0.0.0        240.0.0.0      En vínculo        192.168.186.131  281
 255.255.255.255 255.255.255.255 En vínculo        127.0.0.1     331
 255.255.255.255 255.255.255.255 En vínculo        192.168.186.131  281
=====
Rutas persistentes:
 Ninguno
  En vínculo = no existe puerta de enlace para ese destino
=====
IPv6 Tabla de enrutamiento
=====
Rutas activas:
 Cuando destino de red métrica   Puerta de enlace
 2   331 ::/0                     En vínculo
 1   331 ::1/128                 En vínculo
 2   331 2001:::/32               En vínculo
 2   331 2001:0:9d38:953c:1c6e:1a6d:3f57:457c/128
                                         En vínculo
 4   281 fe80::/64                En vínculo
 2   331 fe80::/64                En vínculo
 2   331 fe80::1c6e:1a6d:3f57:457c/128
```

No es una dir IP válida. Se usa esta ruta cuando no tenemos dirección asignada o cuando no existe ninguna otra ruta para alcanzar un destino (por ejemplo desconocido) y entonces señalamos a la puerta de enlace predeterminada, que en este caso es la del router (192.168.186.2), y para alcanzarlo debemos salir por el interfaz que tiene asignada la IP 192.168.186.131 (el que tiene nuestro equipo)

IP=127.0.0.1 es localhost (dirección de loopback)

Básicamente cada línea nos dice que para alcanzar un determinado destino (IP) debemos enviar los paquetes a través de un interfaz de red (podemos tener varios) y si deben ir dirigidos a una determinada puerta de enlace (por ejemplo porque nosotros no tenemos el destino en nuestras tablas de enrutamiento y "le pasamos la pelota" al Gateway)

## Ejemplo de tabla de enrutamiento en Linux

Para ver (o modificar) la tabla de enrutamiento de nuestro sistema Ubuntu podemos usar el comando "**route**". Te lo mostramos en un ejemplo:

```

usuario@ubuntu:~$ route
Comando "route", nos muestra la tabla de enrutamiento del sistema
Tabla de rutas IP del núcleo
Destino     Pasarela      Genmask        Indic Métric Ref    Uso Interfaz
default     192.168.43.1  0.0.0.0        UG    100    0      0 ens33
Link-local  *             255.255.0.0   U     1000   0      0 ens33
192.168.43.0 *            255.255.255.0  U     100    0      0 ens33
usuario@ubuntu:~$ route -n
máscara de red para la dirección de destino; '255.255.255.255' si el destino es un ordenador y '0.0.0.0' para la ruta por defecto
U (la ruta está activada (up))
H (el objetivo es un ordenador anfitrión (host))
G (usa un gateway)
R (restablece una ruta para encaminamiento dinámico)
D (instalada dinámicamente por un demonio o redirección)
M (modificada a partir del demonio de ruta o redirección)
! (ruta de rechazo)

Tabla de rutas IP del núcleo
Destino     Pasarela      Genmask        Indic Métric Ref    Uso Interfaz
0.0.0.0     192.168.43.1  0.0.0.0        UG    100    0      0 ens33
169.254.0.0 0.0.0.0     255.255.0.0   U     1000   0      0 ens33
192.168.43.0 0.0.0.0     255.255.255.0  U     100    0      0 ens33

```

La opción "-n" nos muestra la información en formato numérico (IP)

máscara de red para la dirección de destino; '255.255.255.255' si el destino es un ordenador y '0.0.0.0' para la ruta por defecto

U (la ruta está activada (up))  
H (el objetivo es un ordenador anfitrión (host))  
G (usa un gateway)  
R (restablece una ruta para encaminamiento dinámico)  
D (instalada dinámicamente por un demonio o redirección)  
M (modificada a partir del demonio de ruta o redirección)  
! (ruta de rechazo)

Normalmente, cuando nos conectamos a Internet a través de un *router*, este tiene dentro de nuestra red una dirección tipo "192.168.1.1" y funciona como puerta de enlace predeterminada (*gateway*) en la configuración IP de nuestro equipo.

Al estar conectado a Internet el router tendrá además una dirección IP pública, que será la de nuestro acceso particular a Internet y la compartiremos todos los equipos conectados internamente en nuestro domicilio (y el *router* se encarga de hacer las traducciones correspondientes - NAT - para que todos "salgamos" a Internet).

Si estamos conectados a través de una red WIFI pública veremos que la dirección IP del *gateway* predeterminado suele ser diferente.

Hay muchas formas de conocer la dirección IP pública que tiene nuestro acceso, por ejemplo en estas páginas:

<http://www.cualesmiip.com/>.

<https://www.adslayuda.com/ip.html>.

## Ejercicio

Una de las formas de especificar la dirección IP de una red es indicando el número de bits que abarca su máscara de subred, por ejemplo:

91.116.124.19 / 27

Lo anterior indicaría que nuestra dirección IP es "91.116.124.19"; es una dirección de red de "CLASE A" y la máscara de subred ocupa 27 bits, es decir, esta máscara sería: "255.255.255.224"

Gráficamente lo podemos representar como:



De lo anterior se deduce que disponemos de **5 bits** para los posibles "*hosts*" dentro de la subred, lo cual daría para 32 posibles equipos, pero como no podemos usar la combinación "00000", reservada para la dirección de la red (91.116.124.0), ni tampoco la combinación "11111", reservada para la dirección de difusión dentro de esa subred (91.116.124.31), entonces solamente podremos incluir 30 equipos ("*hosts*") como máximo con una IP cada uno (o menos si alguno necesita más de una dirección, claro).

Teniendo en cuenta lo anterior, te proponemos el siguiente ejercicio: si tu dirección IP fuese "120.205.36.25 / 25"...

¿Cuál sería el rango de direcciones IP que podríamos asignar a los equipos de la subred en la que nos encontramos?

Aunque te ponemos la respuesta abajo, te pedimos que siguiendo el mismo esquema de razonamiento que acabas de ver, hagas tú el cálculo antes de verlo.

**Respuesta:** te soplamos la respuesta, podríamos asignar direcciones desde la 120.205.36.1 hasta la 120.205.36.126, es decir, podríamos tener 126 direcciones asignadas a los *host* de la subred.

Por si te sirve de ayuda, te recomendamos una aplicación web que te puede ayudar para practicar con las subredes: <http://www.calculadora-redes.com/>

# Gestión de puertos

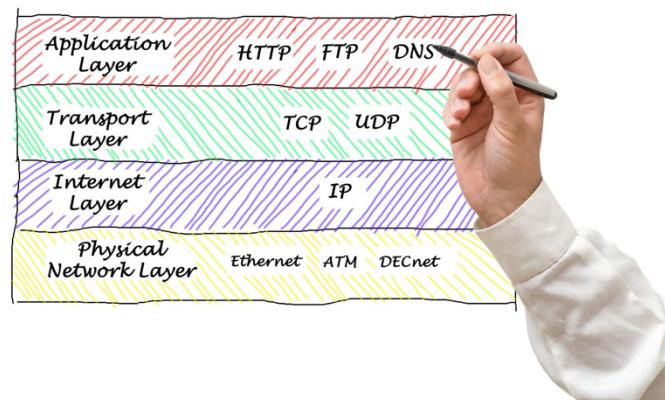
## Protocolos TCP y UDP

Ya conocemos el concepto de puerto y sabemos que, a la hora de configurar el sistema operativo, una buena práctica de seguridad es cerrar los puertos que no estén siendo utilizados, y la conveniencia de monitorizar su estado.

Lo que llamamos **puerto de red** es la identificación de una interfaz de comunicación por encima del nivel de red (en la capa 4 – “nivel de transporte”), que da un determinado servicio de comunicación a un programa (por ejemplo, servicio con conexión o sin conexión).

En este nivel hablamos de “segmentos” para referirnos a las unidades de información a enviar/recibir por la red.

Los protocolos más conocidos que proporcionan el servicio a nivel de capa de transporte, como sabemos, son: **TCP (Transmission Control Protocol)** y **UDP (User Datagram Protocol)**, pero no debemos confundirlos con los puertos (interfaces) sobre los que actúan.



### El protocolo TCP (Transmission Control Protocol)

Este protocolo proporciona un **servicio (orientado a conexión) de transporte fiable de información extremo** a extremo entre aplicaciones. El propio protocolo implementado sobre el puerto se encarga de asegurar la fiabilidad de la transferencia de los datos (encargándose de realizar retransmisiones, gestionar posibles pérdidas de paquetes, restablecer el orden de los paquetes, etc.).

Las comunicaciones realizadas usando protocolo TCP son fiables, pero a la vez el trabajo de implementar las acciones y comprobaciones del protocolo las hace más lentas, y la información de gestión añadida es un coste que hay que asumir en la transmisión.

## El protocolo UDP (User Datagram Protocol)

Se trata de un **protocolo “no orientado a conexión”**, y **no garantiza el transporte fiable de los datos**, simplemente envía a través de la red unidades de información (a las que llamamos **“datagramas”**), y añade muy poca información de gestión (8 bytes, frente a los 20 bytes que añade TCP). Como consecuencia, perdemos en fiabilidad y ganamos en rapidez en el envío de los datos. Un servidor que utilice UDP normalmente puede soportar más clientes que con TCP.

UDP se utiliza, por ejemplo, en algunas utilidades de intercambio de ficheros (como RCP) entre ordenadores, y en la transferencia de vídeo y audio.

## Los puertos de comunicaciones

Los **“puertos”** se identifican por un número (2 bytes), y normalmente cada programa tendrá configurado (o habrá que hacerlo) cuál es el puerto(s) que puede utilizar para el intercambio de información a través de la red.

Si un programa usa varios puertos puede, por ejemplo, intercambiar información desde la misma dirección IP, pero a través de interfaces de comunicación (puertos) distintos.

Los números de puerto (16 bits) van desde el 0 al 65 535, y aunque en principio se podría usar cualquiera de ellos para cualquier protocolo, su asignación está normalizada por el IANA. Suelen dividirse en tres categorías:

- **Puertos bien conocidos (0 al 1023):** son puertos reservados para el S.O., para protocolos conocidos como HTTP, POP3/SMTP o Telnet. Para usar alguno de ellos debemos arrancar el servicio que los usará con permisos de administrador.
- **Puertos registrados (1024 al 49151):** pueden ser usados por cualquier aplicación, y existe una lista pública con los protocolos usados por cada puerto disponible en la web del IANA. Ver: <http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>.
- **Puertos dinámicos o privados (49152 al 65535):** se suelen asignar dinámicamente a las aplicaciones de clientes al iniciarse la conexión. Usados por ejemplo en conexiones “peer to peer” (P2P).

## Los puertos más conocidos

Algunos de los puertos más conocidos son:

- **21:** FTP, usado para la descarga de archivos en el equipo.
- **23:** Telnet, usado para este protocolo de comunicación.
- **25:** SMTP, usado por clientes de email para enviar mensajes de correo electrónico.
- **80:** HTTP, lo usan los navegadores para cargar las páginas web.
- **101:** *hostname*.
- **110 y 995:** puertos POP3, usados por clientes de email para recibir los mensajes de correo electrónico.
- **119:** puerto NNTP.
- **139:** puerto de NetBIOS.
- **443:** HTTPS, usado para la carga segura de páginas web.
- **445:** móvil IP.
- **531:** puerto IRC, usado para servicios de chat.
- **993:** IMAP sobre SSL.
- **995:** POP3 sobre SSL.
- **1521:** puerto usado por Oracle y SQL.
- **3306:** puerto usado por bases de datos MySQL.
- **4661, 4662, 4665:** puertos usados para conexiones *peer to peer* (p. ej. Emule, etc.).

## Gestión de los puertos en Windows y Linux

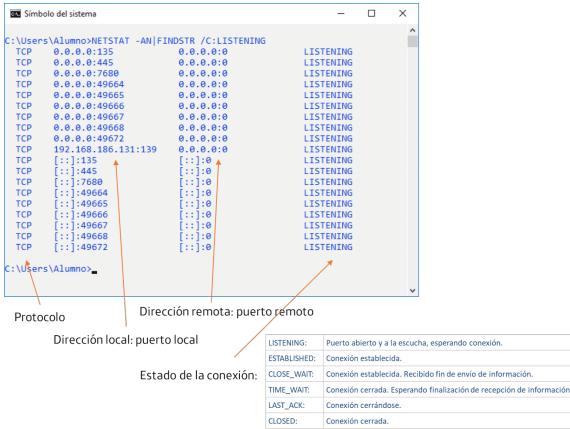
La gestión de los puertos procura que estos se encuentren disponibles para aquellas aplicaciones que necesitan utilizarlos, y al mismo tiempo que no se permita su utilización por parte de programas no autorizados.

Un puerto puede estar **abierto**; en este caso puede ser utilizado por aplicaciones (programas). O bien estar **cerrado** y no podrá utilizarse para el intercambio de información.

Los puertos son administrados por el sistema operativo, pero quien los abre son las aplicaciones que necesitan usarlos. A menudo, para cerrar un puerto es suficiente con cerrar el programa o servicio que lo mantiene abierto. Evidentemente, podemos cerrar “manualmente” aquellos que por seguridad deseemos que no puedan ser utilizados sin el consentimiento del administrador.

Cada sistema operativo incorpora sus propios comandos de gestión de puertos, al igual que los router de cada fabricante.

# Cómo ver los puertos abiertos en Windows



## *Ejemplo de visualización de puertos con netstat.*

En Windows tenemos el comando ***netstat***, con el que podemos conocer qué puertos tenemos abiertos "a la escucha", e incluso cuáles están recibiendo o transmitiendo información con el exterior. Si queremos utilizarla basta con teclear en una consola:

## ***NETSTAT -AN|FINDSTR /C:LISTENING***

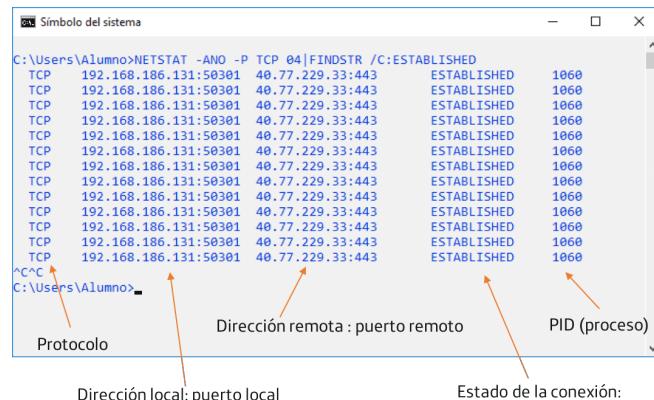
## Cómo ver las conexiones establecidas en Windows

Para ver las conexiones establecidas en un determinado momento y las direcciones IP del destino con el que están conectadas, podemos introducir en la consola:

**NETSTAT -ANO -P TCP  
04|FINDSTR /C:ESTABLISHED**

El comando listará las conexiones “en tiempo real”, y seguirá ejecutándose hasta que terminemos el proceso (p. ej. con “Ctrl+C”).

Este comando nos puede servir para averiguar si nuestro equipo está estableciendo conexiones no deseadas con alguna dirección IP.



# Cómo ver los puertos y conexiones en Linux

```

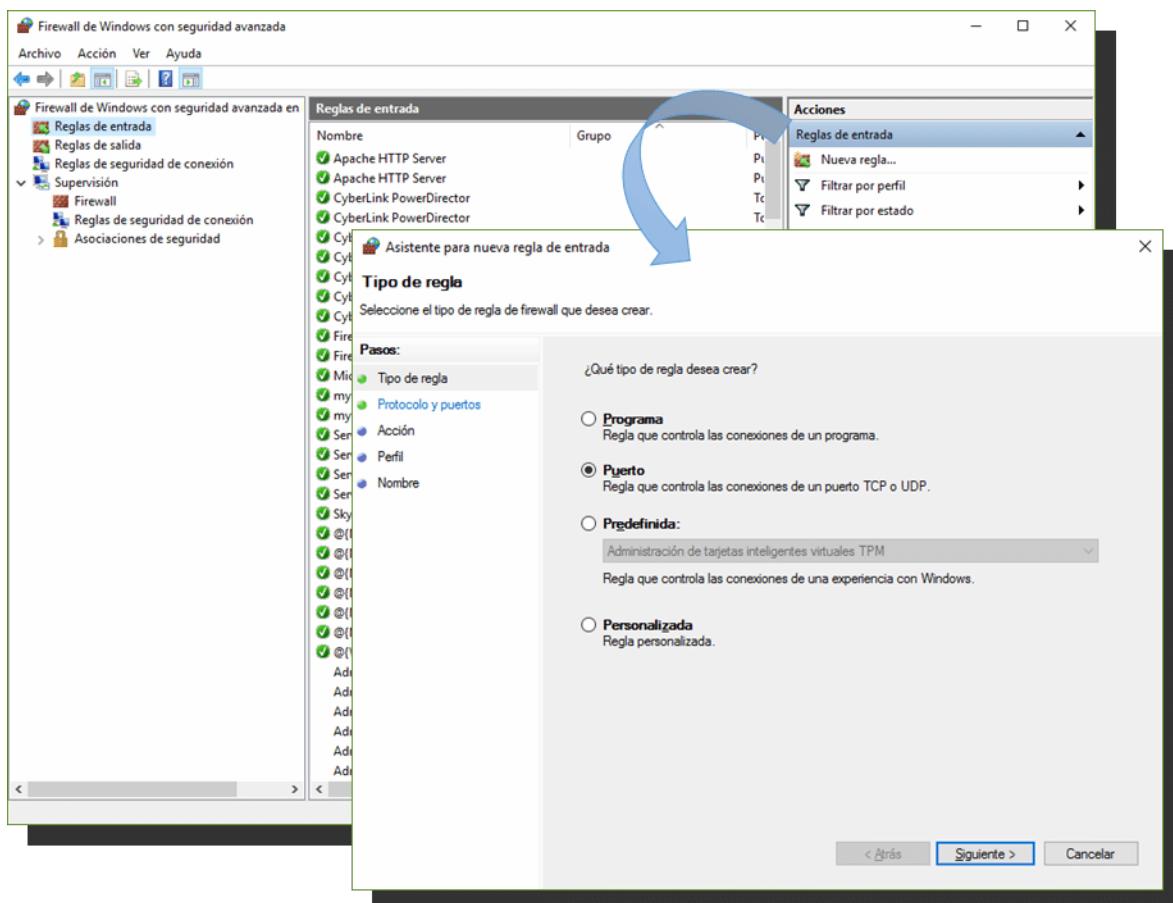
root@pcold:~# sudo netstat -plut
[sudo] password for pcold:
Conexiones activas de Internet (solo servidores)
Proto Recibl Envíad Dirección local           Dirección remota         Estado    PID/Program name
tcp     0      0      localhost:*
```

Para ver los puertos abiertos en Linux (Ubuntu) podemos hacerlo a través de un terminal y con el comando “***netstat***” y algunas de sus opciones.

Por ejemplo, si ejecutamos “`netstat -plut`” como administradores nos devuelve también el nombre del proceso que tiene abierta la conexión.

# Ejemplo de cómo cerrar un puerto en Windows

Aunque cada S.O. tiene su forma determinada de hacerlo, veamos cómo actuar sobre un puerto para cerrarlo utilizando el *firewall* de Windows. Si activamos el *firewall* de Windows y seleccionamos las opciones avanzadas de seguridad, y elegimos crear una nueva regla (por ejemplo para el tráfico de entrada), nos ofrece varias opciones y entre ellas la de hacerla sobre uno de los puertos. Una vez elegida esa opción nos indicará qué tipo de tráfico queremos restringir (si TCP o UDP) y sobre qué número de puerto. Al seleccionarlo y continuar nos saldrán varias opciones para restringir el uso del puerto.



## Algunas recomendaciones

Resulta necesario entonces conocer los comandos de gestión de puertos disponibles en nuestro sistema operativo, y por supuesto también las opciones de configuración a través de la interfaz gráfica. Tengamos en cuenta que un puerto podemos abrirlo/cerrarlo tanto a nivel de S.O. como a nivel de *router* o *firewall*, y este puede estar en el mismo equipo o ser uno independiente.

En general, a modo de recomendaciones podemos tener en cuenta que:

- Disponer de puertos abiertos sin necesidad aumenta considerablemente los riesgos frente a atacantes externos.
- En los *router* los puertos se dirigen a una sola dirección IP, por lo que si tenemos un sistema con varios equipos corriendo la misma aplicación, quizás deberemos configurarla para que use puertos diferentes en cada equipo.
- Si abrimos un puerto en el *router* debemos asegurarnos de que también está abierto en el *firewall* (si existen ambos).
- Si se usa un sistema de asignación dinámica de direcciones (DHCP) y queremos establecer una configuración determinada de puertos asociados a IP concretas, debemos asegurarnos de que esas IP se asignan manualmente y no de forma automática (aunque permanezca el DHCP para la asignación de otras direcciones).
- Lo aconsejable suele ser reservar en el servidor DHCP un rango de direcciones suficiente para los equipos que puedan solicitarla y también reservar un rango de IP para los equipos que necesitan tenerla configurada manualmente.
- La configuración detallada de qué puertos pueden y deben estar abiertos y cuáles deben permanecer cerrados es un elemento esencial de la seguridad de nuestro sistema.

# Despedida

## Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado:

- Resulta importante conocer el formato de las direcciones IP y saber configurar adecuadamente los parámetros de conexión a la red.
- Saber que el procedimiento para configurar los protocolos TCP/IP puede variar según el sistema operativo que estemos empleando, pero los parámetros y direcciones a emplear serán los mismos en un sistema u otro.
- Comprender lo que es la “**máscara de subred**”, una combinación de bits que nos sirve para delimitar el rango de direcciones, es decir, el tamaño de la subred.
- Debemos tener claro además los conceptos de configuración estática y dinámica del cliente de red IP, lo que son los DNS y las tablas de enrutamiento.
- Por último, además de las direcciones a nivel de red, debemos ser conscientes de que los puntos de acceso de las aplicaciones y protocolos de nivel superior se dirigen a través de "puertos" que pueden estar en diversos estados, y que tienen mucho que ver con la seguridad del sistema.