

2º Desarrollo de Aplicaciones Web

IES Doñana

Curso: 2019 - 2020



Despliegue Aplicaciones Web

Tema 2 - Servicio DHCP y DNS



Profesor: Miguel Ángel García

ÍNDICE DE CONTENIDOS

1	<i>Configuración TCP/IP</i>	4
2	<i>Direcciones IP públicas y privadas</i>	4
3	<i>NAT: Traducción de Direcciones de Red</i>	5
3.1	¿Cómo funciona?.....	5
3.2	Ventajas de la NAT	6
4	<i>¿Qué es el servicio DHCP?</i>	7
5	<i>Configuración de los parámetros de red</i>	7
5.1	Parámetros de configuración de red.	7
5.2	Configuración manual de parámetros de red.....	8
5.3	Configuración automática de los parámetros de red.....	8
5.4	Archivo de configuración de parámetros de red.	8
6	<i>El protocolo DHCP</i>	9
6.1	Tipos de asignaciones en DHCP	10
7	<i>Funcionamiento de DHCP</i>	10
7.1	Mensajes DHCP.	11
7.2	Parámetros asignados por DHCP.	13
8	<i>Como configurar el servidor DHCP</i>	13
8.1	Instalación y administración del servicio DHCP - Linux.	13
8.2	Inicio y parada del servicio DHCP.	15
8.3	Archivos relacionados con la configuración y administración de DHCP	16
8.4	El archivo de configuración del servicio DHCP.....	17
8.4.1	Declaraciones.....	17
8.4.2	Parámetros	19
9	<i>El Servicio DNS</i>	22
9.1	Sistemas de nombres planos y jerárquicos.	22
9.2	Espacio de nombres de dominio.	23
9.3	Tipos de dominio.....	23
9.4	Delegación DNS.....	24
9.5	Funcionamiento del servicio DNS.....	24
9.5.1	Proceso de resolución de un nombre de dominio	25
9.5.2	Resoluciones directas y resoluciones inversas.....	25
10	<i>Base de Datos</i>	26

10.1	Estructura	26
10.1.1	Tipos de registros SOA.....	27
10.1.2	Tipos de registros NS	28
10.1.3	Tipos de registros A, PTR, CNAME Y MX.....	28
10.1.4	Tipo de registro SRV.	29
11	<i>Tipos de servidores de DNS.....</i>	29
12	<i>Instalación y configuración del servidor DNS en Linux.</i>	30
12.1	Crear una zona primaria	30
12.2	Crear una zona inversa	31
12.3	Configurar un DNS secundario.	31
12.4	Utilización de reenviadores externos	32
13	<i>Comandos relativos a la resolución de nombres.....</i>	32
13.1	nslookup	33
13.2	Dig.....	33
13.3	Otros comandos.	34

Fuentes usadas para la elaboración del tema:

-  Wikipedia
-  Recursos del IOC

1 Configuración TCP/IP

A estas alturas del ciclo, ya sabrás que para que los PCs de una red puedan intercomunicarse entre sí, deben disponer de una **dirección IP** y de una **máscara de subred**. Además, si queremos que disponga de conexión a Internet, es necesario configurar la **dirección IP de la puerta de enlace** y la **dirección IP de dos servidores DNS**.



Ejemplo configuración TCP/IP

Dirección IP	192.168.0.15
Máscara de subred	255.255.255.0
Puerta de enlace	192.168.0.254
DNS preferido	80.58.0.33
DNS alternativo	80.58.32.97

Dentro de una misma red, los PCs deben tener una dirección IP perteneciente al rango de dicha red. Si el rango es desde 192.168.0.0 hasta 192.168.0.255, las IPs de los PCs deberán tener los tres primeros números iguales (192.168.0.X) y el último número podrá cambiar desde 1 hasta 254, porque no se permite la utilización de la primera ni de la última dirección IP del rango ya que quedan reservadas. Cada PC deberá tener una dirección IP diferente. Si dos PCs tienen la misma IP, habrá un conflicto de IP y ninguno de ellos podrá comunicarse hasta que no se resuelva el conflicto cambiando la IP a uno de ellos. Si no sabemos qué IP poner, podemos ver la IP de otro PC de nuestra red en el que funcione correctamente la conexión de Internet y por regla general, cambiar el último valor por otro diferente que no tenga ningún otro PC.

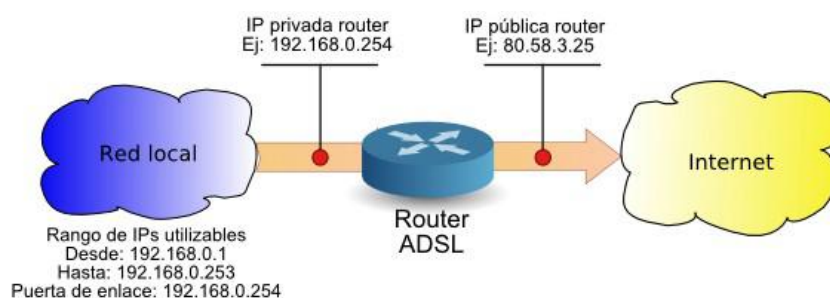
La **Máscara de subred** determina el número de PCs del rango. Casi siempre se suele utilizar la máscara 255.255.255.0 que corresponde a un rango de 256 direcciones IP (suficientes para casi todos los centros educativos) en los que todos los PCs tienen los tres primeros números de la IP iguales y solo cambia el último. Lo normal es que todos los PCs de nuestra red tengan configurada la misma máscara de subred. Si no sabemos cuál es la máscara de subred, podemos verla en otro PC que funcione correctamente la conexión de Internet.

La **Puerta de enlace** deberá ser una IP del rango ya que de lo contrario, nuestro PC no será capaz de comunicarse con ella y no tendrá acceso a Internet. Lo normal es que todos los PCs de nuestra red tengan configurada la misma puerta de enlace. Si no sabemos la IP de nuestra puerta de enlace, podemos verla en otro PC que funcione correctamente la conexión de Internet.

Los **DNS preferido y alternativo** nos los debe proporcionar la compañía que presta el servicio. Telefónica usa el 80.58.0.33 y el 80.58.32.97. Lo normal es que todos los PCs de nuestra red tengan configurados los mismos DNSs. Si no sabemos la IP de los DNS preferido y alternativo, podemos verlos en otro PC que funcione correctamente la conexión de Internet.

2 Direcciones IP públicas y privadas

Las direcciones IP de los PCs de una red local son direcciones privadas ya que los PCs no están directamente conectados a Internet. Solamente el router dispone de conexión directa a Internet y por eso es el único que dispone de una dirección IP pública. Cuando los PCs de una misma red se quieren comunicar unos con otros, lo hacen directamente, pero si quieren comunicarse con Internet, deben hacerlo a través del router. Es equivalente a una



centralita telefónica. Los teléfonos internos de una empresa utilizan números privados (extensiones) y las llamadas al exterior es necesario hacerlas a través de la centralita, que es la única que tiene números de teléfono públicos. Los únicos rangos de direcciones que se se pueden utilizar en redes locales son:

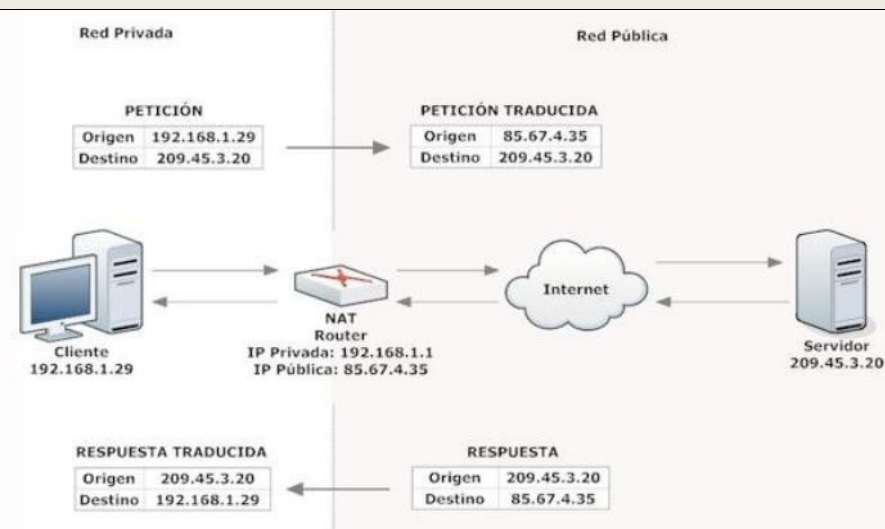
Rangos Redes Locales	
Desde	Hasta
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

Estos rangos de direcciones no están asignados a direcciones públicas de Internet, sino que se han reservado para ser utilizados en las redes locales. Si en lugar de configurar nuestra red con estas direcciones utilizamos otro rango, como seguramente sea un rango utilizado por servidores de Internet, no tendremos acceso a dichos servidores.

3 NAT: Traducción de Direcciones de Red

Internet en sus inicios no fue pensado para ser una red tan extensa, por ese motivo se reservaron “sólo” 32 bits para direcciones, pero el hecho es que el número de máquinas conectadas a Internet aumentó exponencialmente y las direcciones IP se agotaban. Por ello surgió la NAT o Network Address Translation (Traducción de Direcciones de Red). La idea es sencilla, hacer que redes de ordenadores utilicen un rango de direcciones especiales (IPs privadas) y se conecten a Internet usando una única dirección IP (IP pública). Gracias a este “parche”, las grandes empresas sólo utilizarían una dirección IP y no tantas como máquinas hubiese en dicha empresa. También se utiliza para conectar redes domésticas a Internet.

La idea básica que hay detrás de NAT es traducir las IPs privadas de la red en una IP pública para que la red pueda enviar paquetes al exterior; y traducir luego esa IP pública, de nuevo a la IP privada del PC que envió el paquete, para que pueda recibirlo una vez llega la respuesta.

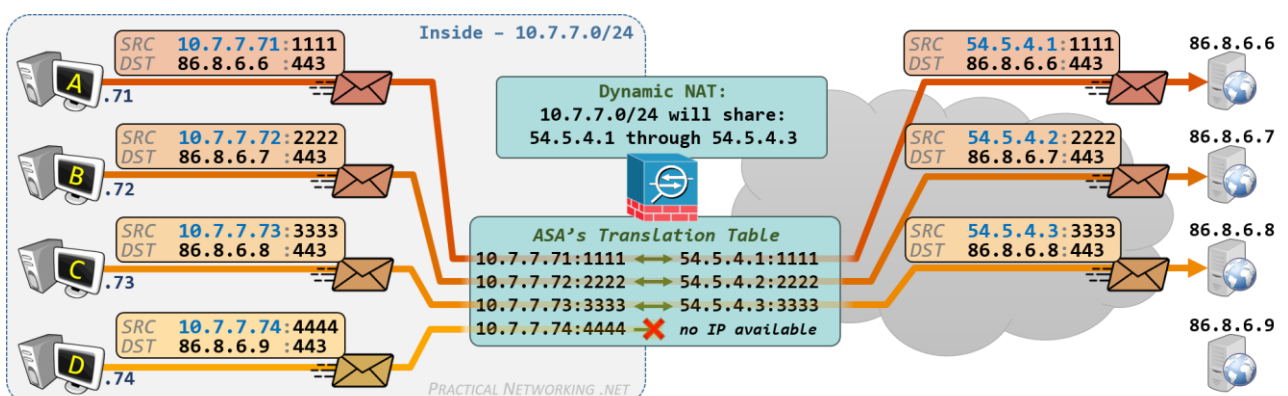


3.1 ¿Cómo funciona?

En la NAT existen varios tipos de funcionamiento:

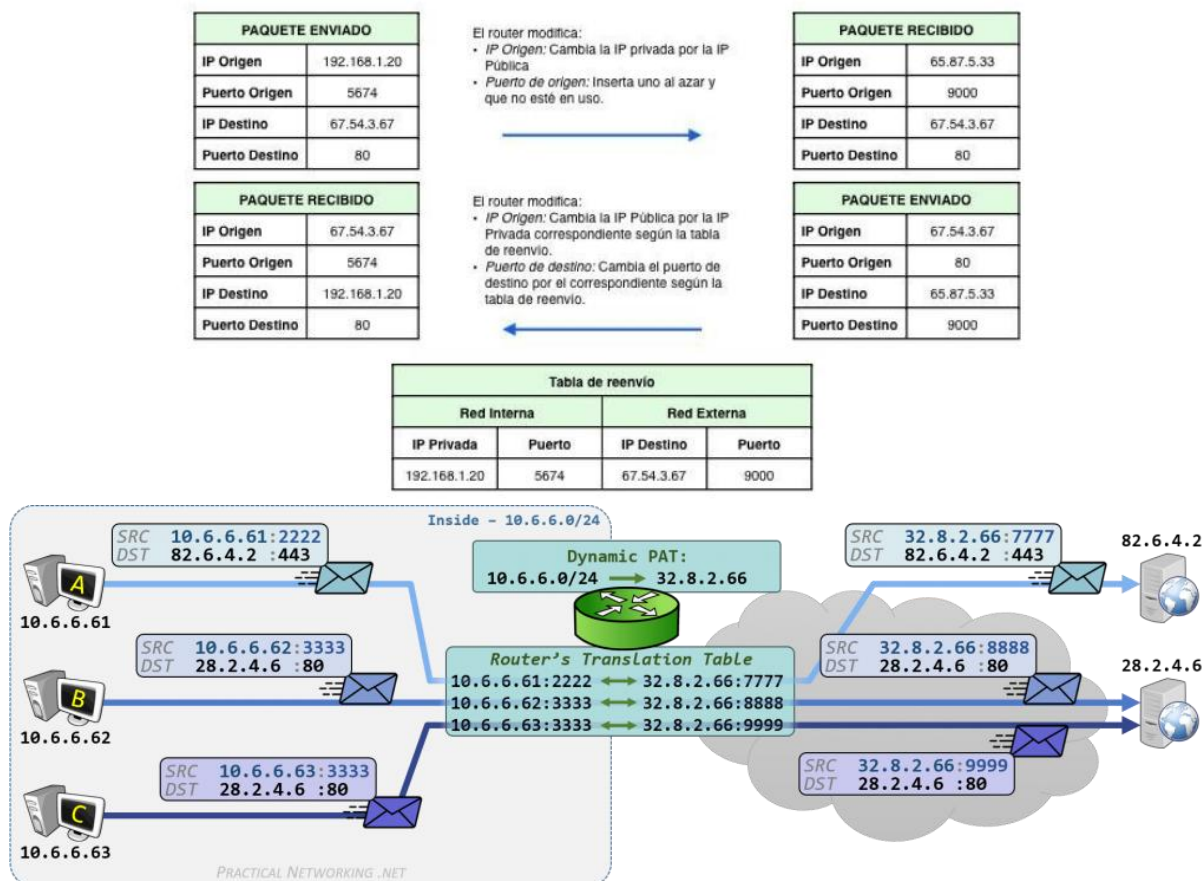
- ✚ **Estática:** Una dirección IP privada se traduce siempre en una misma dirección IP pública. Este modo de funcionamiento permitiría a un host dentro de la red ser visible desde Internet. (Ver imagen anterior)
- ✚ **Dinámica:** El router tiene asignadas **varias direcciones IP públicas**, de modo que cada dirección IP privada se mapea usando una de las direcciones IP públicas que el router tiene asignadas, de modo que a cada dirección IP privada le corresponde al menos una dirección IP pública.

Cada vez que un host requiera una conexión a Internet, el router le asignará una dirección IP pública que no esté siendo utilizada. En esta ocasión se aumenta la seguridad ya que dificulta que un host externo ingrese a la red ya que las direcciones IP públicas van cambiando.



- ✚ **Sobrecarga:** La NAT con sobrecarga o PAT (Port Address Translation) es el más común de todos los tipos, ya que es el utilizado en los hogares. Se pueden mapear múltiples direcciones IP privadas a través de una dirección IP pública, con lo que evitamos contratar más de una dirección IP pública. Además del ahorro económico, también se ahorran direcciones IPv4, ya que aunque la subred tenga muchas máquinas, todas salen a Internet a través de una misma dirección IP pública.

Para poder hacer esto **el router hace uso de los puertos**. En los protocolos TCP y UDP se disponen de 65.536 puertos para establecer conexiones. De modo que cuando una máquina quiere establecer una conexión, el router guarda su IP privada y el puerto de origen y los asocia a la IP pública y un puerto al azar. Cuando llega información a este puerto elegido al azar, el router comprueba la tabla y lo reenvía a la IP privada y puerto que correspondan.



3.2 Ventajas de la NAT

El uso de la NAT tiene varias ventajas:

- ✚ La primera y más obvia, el **gran ahorro de direcciones IPv4** que supone, recordemos que podemos conectar múltiples máquinas de una red a Internet usando una única dirección IP pública.
- ✚ **Seguridad.** Las máquinas conectadas a la red mediante NAT no son visibles desde el exterior, por lo que un atacante externo no podría averiguar si una máquina está conectada o no a la red.
- ✚ **Mantenimiento de la red.** Sólo sería necesario modificar la tabla de reenvío de un router para desviar todo el tráfico hacia otra máquina mientras se llevan a cabo tareas de mantenimiento.

Recordemos que la NAT es solo un parche, no una solución al verdadero problema, por tanto también tiene una serie de desventajas asociadas a su uso:

- ✚ **Checksums TCP y UDP:** El router tiene que volver a calcular el checksum de cada paquete que modifica. Por lo que se necesita mayor potencia de computación.





No todas las aplicaciones y protocolos son compatibles con NAT. Hay protocolos que introducen el puerto de origen dentro de la zona de datos de un paquete, por lo que el router no lo modifica y la aplicación no funciona correctamente.

4 ¿Qué es el servicio DHCP?

Voy a explicarte en qué consiste este servicio. Todos los nodos de una red deben tener una dirección IP, esta se puede asignar de forma manual o automática. El protocolo DHCP (Dynamic Host Configuration Protocol) o Protocolo de Configuración Dinámica de Host, es un protocolo que permite a los nodos de una red obtener su dirección IP de forma automática.

DHCP es un **protocolo** de tipo **cliente/servidor** en el que un servidor tiene rangos de direcciones IP y las va asignando a los clientes según van estando libres. El servidor sabe en todo momento quien tiene la posesión de una IP mediante la dirección MAC

Al usar este protocolo podrás comprobar las siguientes **ventajas**:





-  No se necesita apuntar la configuración de los equipos.
-  Se protegen las IP de los servidores.
-  No hay conflictos de IP.
-  Se pueden reutilizar las direcciones IP.

5 Configuración de los parámetros de red.

En esta unidad vas a conocer el funcionamiento de un **servicio de configuración automática de los parámetros de conexión en red** y como se instala, configura y prueba el servicio.

En el módulo “**Sistemas Informáticos**” de primer curso del ciclo que estás estudiando se trató la arquitectura TCP/IP y, dentro de ella, el protocolo IP y el sistema de direccionamiento IP usado para asignar direcciones a las interfaces de red de los equipos (ordenadores, routers, impresoras de red y otros). Un ordenador va a tener en cada adaptador o tarjeta de red una configuración de parámetros de red.

Los parámetros de red que principalmente se configuran en un adaptador de red sirven para que:

-  Un ordenador tenga la dirección IP y máscara adecuada para conectarse en red a través del adaptador de red.
-  Se pueda conocer si una dirección IP con la que se conecte está en la misma red que el adaptador.
-  Un ordenador se pueda conectar con otras redes a través de un dispositivo enrutador.
-  Un servidor DNS resuelva los nombres DNS de recursos cuya identificación escribamos en los programas de red mediante un nombre DNS.

La configuración de los parámetros de red de un adaptador de red se puede realizar en cualquier sistema operativo al menos de dos formas: **manual y automática**.



La **configuración manual** también se conoce como **configuración fija o estática**. En esta configuración, el administrador de red se tiene que encargar de escribir en un programa o en un archivo de configuración los valores de los parámetros de configuración. En la **configuración automática**, el administrador simplemente se limita a especificar en los ordenadores que la utilicen, que van a usar esa configuración sin tener que aportar más datos.

5.1 Parámetros de configuración de red.

Para que un adaptador de red tenga conectividad dentro de una red debe tener asignada una **dirección IP** que le identifique y una **máscara de red**. La IP debe pertenecer a la misma red que el resto de equipos de la red. Debes recordar que la máscara junto con la IP permiten determinar si otras direcciones IP con las que se conecte desde el adaptador pertenecen o no a la misma red.



En esta unidad se hará referencia al direccionamiento **IPv4** aunque para **IPv6**, con sus particularidades, se podrá aplicar lo mismo que para IPv4. En IPv4, una dirección IP es un conjunto de 32 bits que representamos con 4 números decimales separados por puntos. La máscara es también un número de 32 bits cuya primera parte es una secuencia de unos e indica que bits de la IP representan la red a la que pertenece esa IP y cuya segunda parte es una secuencia de ceros que indican que bits de la IP identifican al adaptador de red dentro de la red. La máscara se puede representar, al igual que la IP, con 4 números decimales separados por puntos o con el formato /num, donde num indica cuantos bits sirven para identificar la red a la que pertenece una IP.

Por ejemplo, si un adaptador tiene IP 192.168.1.7 y máscara 255.255.255.0, se puede decir también que el adaptador tiene la IP 192.168.1.7/24 indicando que hay 24 unos en la máscara. Por tanto en esa IP el identificador de red es 192.168.1 y el del adaptador dentro de la red es 7.

Para que un ordenador tenga, a través de un adaptador de red, **conectividad con otras redes** y, en particular, con Internet se debe asignar en el adaptador la **IP de la puerta de enlace** o la IP del dispositivo de red que realiza el encaminamiento hacia otras redes. Lo normal es que esta IP sea la interna del router de conexión a Internet. Para que en los programas de red se puedan usar nombres DNS de equipos en lugar de direcciones IP, se deben asignar las **IP de los servidores DNS**.

La puerta de enlace debe pertenecer a nuestra red. Un servidor DNS puede pertenecer o no a nuestra red. Al final de esta unidad aprenderás a instalar y configurar un servidor DNS.

5.2 Configuración manual de parámetros de red.

Al asignar las direcciones IP de forma manual hay que ser cuidadoso y ordenado. Se debe tener documentada la asignación de IP que se ha hecho en cada ordenador de la red para conocer las IP que ya tenemos utilizadas y las que podemos usar. Todas las direcciones IP de los equipos de una red deben pertenecer a la misma IP de red. Las direcciones IP que se asignen se deben elegir de acuerdo a un criterio de ordenamiento y organización de los ordenadores dentro de la red.

Si se asigna la misma IP a dos ordenadores de una red y ambos están en funcionamiento se produce un **conflicto de IP** lo que provoca que esos dos ordenadores no funcionen correctamente en la red.



En los sistemas Linux Debian/Ubuntu se tiene instalado el software de detección y configuración NetworkManager que incluye una herramienta gráfica para configurar conexiones de red.

5.3 Configuración automática de los parámetros de red.

Establecer la **configuración automática** de los parámetros de red para un adaptador de red es muy sencillo y no se necesita conocer nada acerca de esos parámetros. Sin embargo, para que se pueda establecer la configuración automática se necesita tener en la red algún dispositivo que proporcione el servicio de asignación automática de parámetros de red (**servidor**).

Cuando un ordenador cliente se inicia y está configurado para recibir automáticamente los parámetros de red, realiza un proceso para buscar un servidor en la red que le envíe esos parámetros. Si hay algún servidor, el ordenador cliente recibirá los valores de los parámetros de configuración de red desde el servidor, los asignará y podrá trabajar en red sin ningún problema.

Supongo que ante esta explicación tan simple de una asignación automática de parámetros de red, te surgirán muchas preguntas y dudas sobre el funcionamiento, pero no te preocupes, encontrarás respuestas en los contenidos de esta unidad.

La asignación automática de los parámetros de red presenta varias ventajas frente a la asignación manual:

- ✚ Se puede mover un ordenador entre dos redes que dispongan del servicio de asignación automática sin tener que cambiar su configuración.
- ✚ Es mucho más fácil configurar la red en los ordenadores ya que no hay que recordar los parámetros de configuración necesarios ni hay que escribirlos.
- ✚ Se evitan posibles conflictos de IP en la red, ya que en los servidores se controla que se puedan dar dos IP iguales en la red.
- ✚ Se reduce el trabajo dedicado por los administradores de la red a la configuración de la conexión de red de los equipos.

5.4 Archivo de configuración de parámetros de red.

En los sistemas Linux, si no se dispone de una herramienta gráfica de configuración de los parámetros de red, hay que realizar la configuración editando un archivo de configuración de dichos parámetros. En los sistemas Linux Debian/Ubuntu este archivo es **/etc/network/interfaces**. (Para las nuevas versiones de Ubuntu, ver el documento: "Tema 02 - Actividad 5 - Configurar Red - Ubuntu 18.04")

Si queremos que el adaptador de red de nuestro ordenador reciba los parámetros de red de forma automática, este archivo debe contener lo siguiente:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
```


Las dos primeras líneas establecen la configuración de red del adaptador de bucle local. Las dos siguientes son las que establecen la configuración del adaptador "eth0" de conexión a red. En esa configuración, "auto" implica que el adaptador se active en el inicio del sistema e "iface eth0 inet dhcp" significa que el interface o adaptador eth0 tenga una configuración automática de parámetros de red usando el protocolo dhcp.

Si queremos establecer de forma estática la configuración de los parámetros de red del adaptador de red eth0, el archivo debería contener algo como lo siguiente:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.254
dns-nameservers 8.8.8.8
```

En este caso vemos que la configuración que se establece para eth0 es static, es decir, que se asignan manualmente los parámetros que se escriben a continuación en el propio archivo. En el ejemplo se asigna la IP 192.168.1.1 con la máscara de red 255.255.255.0, la dirección de red es 192.168.1.0, la dirección de broadcast es 192.168.1.255, la dirección IP de la puerta de enlace es 192.168.1.254 y el servidor DNS es 8.8.8.8.

A continuación se indican algunos tutoriales para aprender a configurar los parámetros de red en distintos Sistemas Operativos:

LINUX

Configurar la red en la interfaz gráfica de Ubuntu 16.04 LTS - <https://goo.gl/5cFkSk>

Averiguar la IP en un ordenador con Ubuntu 14.04 LTS - <https://goo.gl/XB7b4d>

Consultar la configuración de la red en Ubuntu con ifconfig - <https://goo.gl/mpcA5a>

Configurar la red en Ubuntu modificando el archivo de configuración - <https://goo.gl/iey48X>

Configurar la red en Ubuntu modificando el archivo de configuración (Posterior a Ubuntu Server 17.04)
<https://xurl.es/tu0qw>

Cambiar nombre interface enp0s3 a eth0 en Ubuntu 16.04.1 - <https://youtu.be/RwgYEWqVxzs>

WINDOWS

Asignar una dirección IP fija en Windows 8.1 - <https://goo.gl/22v8Sd>

Averiguar la IP en un ordenador con Windows 10 - <https://goo.gl/JQBB1U>

Consultar la configuración de la red en Windows con ipconfig - <https://goo.gl/vo7apA>

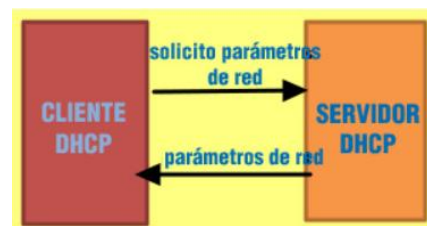
Configurar una dirección IP estática en Windows 10 (VÍDEO) - <https://youtu.be/mBZVp9hGpYw>

6 El protocolo DHCP

El **protocolo DHCP (Dynamic Host Configuration Protocol)**, es un protocolo que corresponde al nivel de aplicación para redes TCP/IP. El objetivo de este protocolo es proporcionar un servicio de red que permita a equipos de una red TCP/IP obtener automáticamente los parámetros de configuración de red.

El funcionamiento de DHCP en la red se basa en un modelo cliente/servidor con uno o varios equipos servidores que proporcionan el servicio y varios equipos clientes que reciben automáticamente los parámetros de configuración de red.

Un cliente DHCP solicita en la red que algún servidor DHCP le proporcione los parámetros de configuración de red. Si hay algún servidor DHCP, el cliente recibirá valores de los parámetros desde el servidor y asignará esos valores para poder empezar a trabajar en red. Un servidor DHCP, aparte de atender a los clientes enviándoles los parámetros, controla cuales son las direcciones IP que ha asignado y cuales están disponibles.



Un servidor DHCP envía, o puede enviar, varios parámetros de red que permitan a los clientes trabajar en red y utilizar todos los servicios disponibles. En explicaciones posteriores, centraremos las cuestiones sobre el funcionamiento DHCP en la asignación de la IP, aunque realmente se asignan más parámetros.

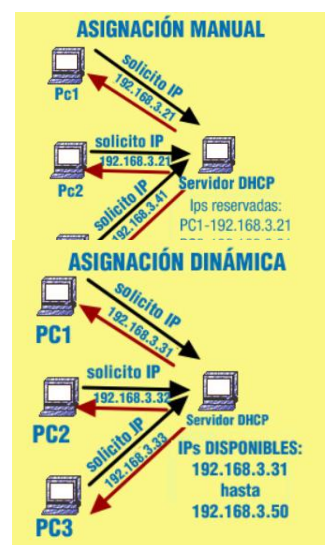
6.1 Tipos de asignaciones en DHCP

Ya sabemos que un cliente DHCP recibe desde un servidor los parámetros de configuración de red y entre ellos la IP. El servidor, por tanto, es quien asigna IP a los clientes DHCP. Pero, ¿cómo decide el servidor la IP que asigna a un cliente? ¿Durante cuánto tiempo asigna una IP a cada cliente? Estas cuestiones se responden en función del tipo de asignación utilizado en el servidor DHCP.

El **tipo de asignación DHCP** es el mecanismo por el cual un servidor DHCP decide la IP que tiene que entregar a un cliente y durante cuánto tiempo concede una licencia o autorización al cliente para que use esa IP.

La IP que asigna un servidor DHCP a un cliente puede ser elegida dentro de un conjunto posible de direcciones IP, o puede ser obligatoriamente una IP concreta. Cuando un cliente DHCP recibe de un servidor una IP para que la utilice, se dice que recibe una **concesión**. La concesión puede asignarse por tiempo limitado o por tiempo ilimitado. El protocolo DHCP establece la posibilidad de utilizar en los servidores DHCP tres técnicas de asignación DHCP:

- ✚ **Asignación estática o manual:** Mediante esta técnica se reserva una IP en exclusiva para un cliente. Un cliente recibe siempre la misma IP. El servidor asocia la IP con una identificación del cliente, que normalmente es la **dirección física del adaptador de red (MAC)**. Siempre que un ordenador cliente solicita al servidor una IP, enviará un identificador y este identificador permitirá al servidor concederle la IP asociada con ese identificador.
- ✚ **Asignación automática:** Mediante esta técnica el servidor DHCP asigna a cualquier cliente DHCP que lo solicite una dirección IP (dentro de todas las que tenga disponibles para conceder) **de forma permanente**. El cliente DHCP va a mantener esa dirección IP mientras no renuncie a ella, es decir, mientras no envíe un mensaje al servidor indicando esa renuncia. La asignación automática tiene como principal problema que si un equipo cliente ha recibido una IP por ese mecanismo, la IP que ha recibido no va a poder ser usada por ningún otro cliente aunque el primero estuviera apagado, incluso por mucho tiempo.
- ✚ **Asignación dinámica:** Mediante esta técnica se asigna a cada cliente DHCP una IP **durante un intervalo de tiempo limitado**. Durante ese tiempo, el servidor DHCP no va a conceder la IP asignada a ningún otro cliente. Para que un cliente pueda mantener una IP previamente concedida, debe renovar la concesión con el servidor antes de que termine el tiempo de concesión. Si termina un tiempo de concesión sin haber hecho la renovación (por ejemplo cuando un ordenador se apaga por tiempo mayor), la IP correspondiente va a poder ser entregada a otro cliente DHCP.



7 Funcionamiento de DHCP

En este apartado vamos a ver todo lo que ocurre cuando un cliente DHCP solicita que se le asigne desde un servidor una IP de forma automática.

Las conexiones DHCP se desarrollan sobre el protocolo de transporte UDP, es decir, los mensajes DHCP que se intercambian un cliente y un servidor son transportados en la red por el protocolo UDP. Un **servidor DHCP usa el puerto UDP 67** mientras que un **cliente DHCP usa el puerto UDP 68**.

Básicamente, el proceso de configuración automática de IP en un cliente DHCP se desarrolla en cuatro pasos:

- ✚ La **solicitud**: el cliente DHCP envía un mensaje de difusión a todos los equipos de la red, en el que solicita que algún servidor DHCP le envíe los parámetros de configuración de red y, fundamentalmente, una IP. En este mensaje, el cliente envía su dirección MAC para que pueda responderle cualquier servidor DHCP que reciba la solicitud.

- ✚ La **Propuesta**: cualquier servidor DHCP de la red que haya recibido la solicitud, envía un mensaje al cliente en el que propone una IP dentro de un rango de IP disponibles para asignar y una máscara. Un servidor DHCP, antes de enviar la propuesta, comprueba si la IP que va a proponer ya se está usando en la red. Para ello envía un mensaje ARP para que le responda algún ordenador que pudiera tener esa IP. Si la IP estuviera ya utilizada, el servidor buscará otra IP para asignar.
- ✚ La **aceptación**: comienza cuando un cliente DHCP ha recibido la propuesta de una IP desde un servidor. Al recibir la propuesta, el cliente comprueba si la IP está siendo usada por algún ordenador de la red, al igual que hizo el servidor. Si la IP no está siendo usada, el cliente envía un mensaje de difusión a toda la red indicando que acepta la propuesta. En el mensaje de difusión indica cual es la IP que acepta y el servidor desde el que se envió la propuesta aceptada. Si el cliente DHCP recibe a continuación otros mensajes de propuesta desde otros servidores, no responde a esos mensajes. El mensaje de aceptación de IP sirve para que todos los servidores DHCP de la red conozcan que el cliente ya ha aceptado una propuesta hecha desde un determinado servidor.
- ✚ La **Confirmación**. el servidor cuya propuesta ha sido aceptada, recibe el mensaje de aceptación y envía al cliente DHCP un mensaje de confirmación que incluye la IP que se asigna, la máscara, otros parámetros de red y otras opciones de configuración. Desde este momento, el cliente dispone de un tiempo de licencia o de concesión para usar la IP. Cuando el cliente ha utilizado la mitad del tiempo de concesión, inicia un proceso de renovación en el que solicita al servidor que le había asignado la IP que le renueve la licencia, es decir, que le conceda seguir usando la IP por el tiempo de concesión.

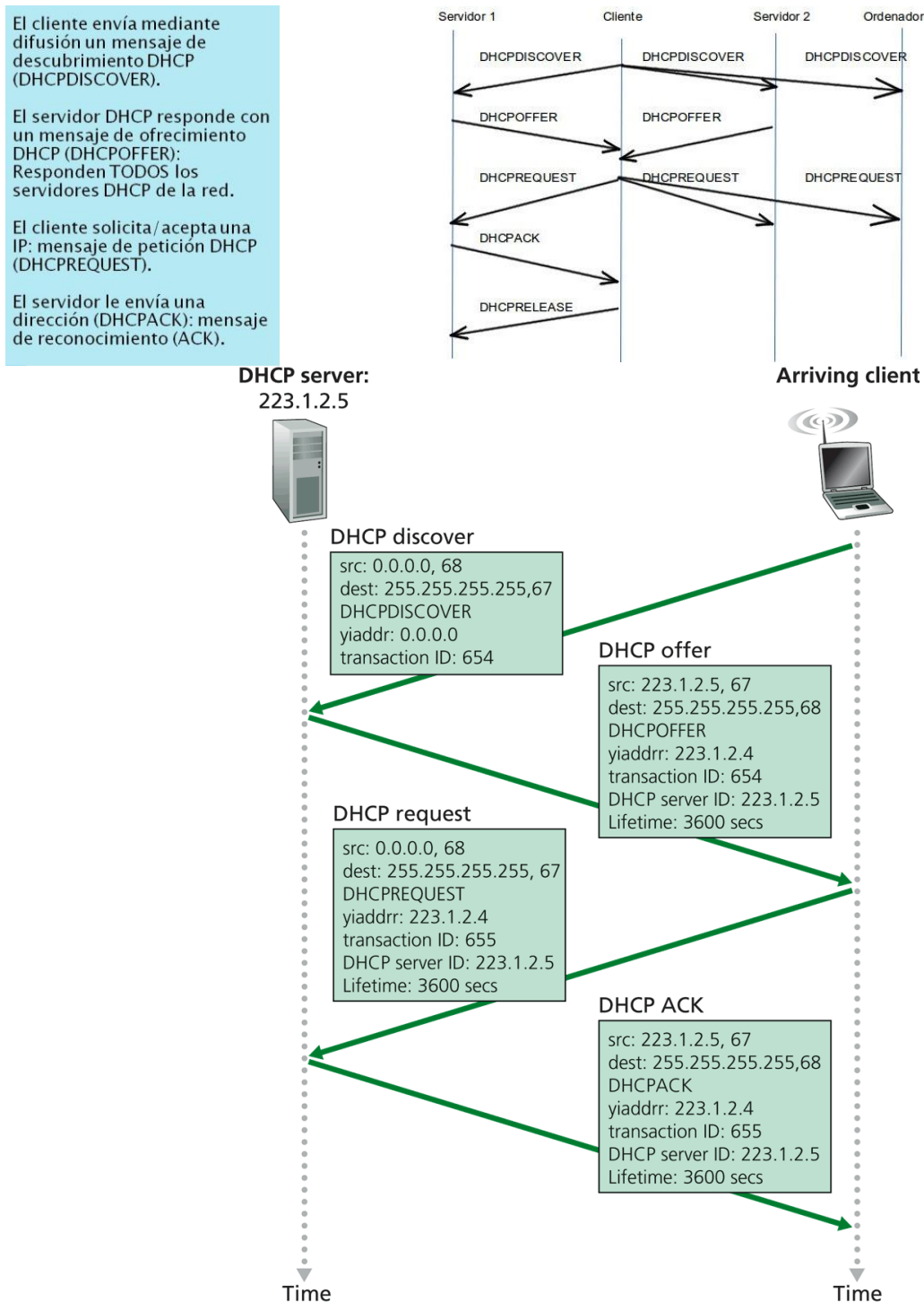
7.1 Mensajes DHCP.

El protocolo DHCP establece cuales son los mensajes que se pueden enviar entre clientes y servidores DHCP y cuál es el formato y significado de esos mensajes. Estos mensajes DHCP son:

- ✚ **DHCP_DISCOVER**: Es enviado por un cliente DHCP para solicitar que algún servidor DHCP de la red le envíe los parámetros de configuración de red. El mensaje es de broadcast o difusión por lo que llegará a todos los posibles servidores DHCP de la red.
- ✚ **DHCP_OFFER**: Es enviado por un servidor DHCP en respuesta a un mensaje DHCP_DISCOVER de un cliente. El servidor ofrece en el mensaje una concesión de una IP junto con valores para los parámetros solicitados por el cliente. Antes de enviar un DHCP_OFFER, el servidor hace una comprobación de que la IP que va a ofrecer no está siendo usada en la red. Si se usa asignación manual o, estática, en el servidor, éste asigna al cliente la IP que tiene reservada para el cliente.
- ✚ **DHCP_REQUEST**: Es un mensaje que se envía desde un cliente en respuesta a un mensaje DHCP_OFFER recibido desde un servidor o bien cada vez que el cliente tiene que renovar una concesión. Mediante este mensaje, el cliente indica al servidor que acepta la oferta hecha por el servidor y solicita que le otorgue una concesión de la IP. Antes de enviar esta respuesta, el cliente comprueba si la IP que se le ha ofrecido está siendo usada ya en la red. Este mensaje es de difusión, lo que sirve para que otros posibles servidores de la red conozcan que el cliente ya ha aceptado una oferta. Si el cliente recibe otras ofertas desde otros servidores no envía respuestas a ellas.
- ✚ **DHCP_ACKnowledge**: Es un mensaje que envía un servidor a un cliente en respuesta a un mensaje DHCP_REQUEST. En este mensaje, el servidor indica al cliente que le asigna la IP solicitada durante un tiempo de concesión establecido. En este mensaje se incluyen valores para el resto de parámetros de configuración. Cuando el cliente recibe este mensaje establece los parámetros de configuración de red con los valores entregados desde el servidor.
- ✚ **DHCP_NAK**: Es un mensaje que se podría enviar desde un servidor a un cliente en respuesta a un mensaje DHCP_REQUEST para indicarle que no puede entregarle la IP que ha solicitado en ese mensaje. No es muy normal el envío de este mensaje. Podría darse en procesos de renovación de concesiones cuando la IP que está solicitando renovar el cliente se ha reservado o está fuera del ámbito de direcciones asignables por el servidor.

- ✚ **DHCP_DECLINE:** Es un mensaje que enviará el cliente DHCP en sustitución de un mensaje DHCP_REQUEST cuando detecta que la IP que se le ha ofrecido ya está siendo usada en la red.
- ✚ **DHCP_RELEASE:** Es un mensaje que envía el cliente DHCP al servidor para indicarle que da por terminada la concesión. Este mensaje no tiene que enviarle un cliente de forma obligatoria cuando desea cancelar una concesión. Si el servidor recibe este mensaje, considera liberada la IP sobre la que el cliente tenía la concesión.
- ✚ **DHCP_INFORM:** Es un mensaje que puede enviarle el cliente DHCP al servidor para solicitarle parámetros adicionales de configuración de red (no recibidos con anterioridad o recibidos y solicitando una actualización).

En las siguientes imágenes se muestra el esquema de intercambio de mensajes DHCP en un proceso de configuración automática de parámetros de red.



7.2 Parámetros asignados por DHCP.

¿Conoces todos los parámetros de configuración de red que se pueden asignar mediante DHCP a un cliente? ¿Sabes cuáles de esos parámetros se asignan de forma obligatoria y cuáles de forma opcional?

Un cliente DHCP puede recibir de un servidor varios **parámetros** de configuración de red. Algunos de esos parámetros son asignados siempre desde el servidor por lo que se consideran **obligatorios**. El resto de parámetros deben ser solicitados por el cliente y pueden ser asignados opcionalmente por el servidor y se consideran **opcionales**.

Parámetros obligatorios:

- ✚ Dirección IP del cliente.
- ✚ Máscara de subred del cliente.
- ✚ Tiempo de concesión o duración de la licencia (lease time).
- ✚ Tiempo de renovación de la licencia (renewal time).
- ✚ Tiempo de reconexión (rebinding time).



Si un cliente agota el tiempo de concesión sin renovar dicha concesión, el servidor considera liberada la IP que ha concedido al cliente.

Un cliente DHCP realiza un proceso de renovación de una concesión para poder seguir usando una IP concedida anteriormente durante un nuevo tiempo de concesión. De forma general, los servidores DHCP asignan un tiempo de renovación igual a la mitad del tiempo de concesión. Un proceso de renovación se inicia con un mensaje DHCP-REQUEST. El tiempo de reconexión especifica que transcurrido ese tiempo sobre una concesión, si no se ha realizado una renovación, habrá que solicitar una IP mediante una nueva conexión DHCP. En esa conexión se podrá recibir la concesión de la IP anterior siempre que no se haya agotado el tiempo de concesión.

Los parámetros opcionales, se definen dentro de las opciones del protocolo DHCP en el RFC 2132. De todos ellos, algunos destacables que se le pueden entregar a los clientes son:

- ✚ Dirección IP de la puerta de enlace.
- ✚ Servidores DNS.
- ✚ Nombre de dominio DNS.
- ✚ Dirección de broadcast en la red.
- ✚ Servidores SMTP (de transferencia de correo).
- ✚ Nombre del servidor WINS.
- ✚ MTU o longitud máxima de la unidad de transferencia para el adaptador de red.
- ✚ Tiempo máximo de espera de respuesta para el protocolo ARP.

8 Como configurar el servidor DHCP.

Ahora llega lo más importante ¿Cómo configuras los servidores? A continuación verás como configurarlo en Ubuntu Server.

Como futuro técnico es conveniente que sepas, que también se puede configurar un router como servidor DHCP. Puede ser incluso que por defecto esté configurado como tal.



8.1 Instalación y administración del servicio DHCP - Linux.

ISC desarrolla de forma oficial software libre DHCP para sistemas Linux. En concreto, desarrolla tres paquetes de software:

- ✚ Software servidor DHCP.
- ✚ Software cliente DHCP.
- ✚ Software agente relay DHCP o agente de Retransmisión de DHCP.

Desde ahora tendrás en cuenta que en este curso haremos principalmente referencia a servicios instalados sobre las distribuciones Ubuntu de Linux. Lo que se indique como válido para Ubuntu, en general, lo será también para distribuciones Debian. Otras distribuciones como CentOS, Fedora, RedHat o como OpenSuse presentan sus particularidades..



Vamos a comenzar viendo un listado de paquetes que contienen la cadena dhcp en el nombre:

```
#apt-cache search -names-only dhcp
```



```
alumno-VirtualBox alumno # apt-cache search --names-only dhcp
maas-dhcp - MAAS DHCP server
neutron-dhcp-agent - Neutron is a virtual network service for Openstack - DHCP agent
strongswan-plugin-dhcp - strongSwan plugin for forwarding DHCP request to a server
isc-dhcp-client - Cliente de DHCP del ISC
isc-dhcp-client-dbg - Cliente ISC DHCP (símbolos de depuración)
isc-dhcp-common - Archivos comunes que utilizan todos los paquetes isc-dhcp*
isc-dhcp-dev - API para el acceso y la modificación del estado del servidor y el cliente de DHCP
isc-dhcp-relay-dbg - Demonio de retransmisión DHCP (símbolos de depuración)
isc-dhcp-server - Servidor ISC DHCP para asignación automática de direcciones IP
isc-dhcp-server-dbg - Servidor ISC DHCP para asignación automática de direcciones IP (depuración)
udhcp - Contiene la implementación del cliente busybox DHCP
dhcp-probe - network DHCP or BOOTP server discover
dhcpd-dbus - Dbus bindings for dhcpd
dhcpd-gtk - GTK+ frontend for dhcpd and wpa supplicant
dhcpd5 - DHCPv4, IPv6RA and DHCPv6 client with IPv4LL support
```

Antes de instalar el dhcp server, comprobaremos que no está instalado.

```
#dpkg -l "*dhcp*"
```

```
alumno-VirtualBox alumno # dpkg -l "*dhcp*"
Deseado=Desconocido/Instalar/Eliminar/Purgar/Retener
| Estado=No/Instalado/Config-files/Desempaquetado/Medio-conf/Medio-inst/espera-disparo/pendiente-disparo
|/ Err?=(ninguno)/Requiere-reinst (Estado,Err: mayúsc.=malo)
||/ Nombre Versión Arquitectura Descripción
+++-----+-----+-----+-----+
un dhcp-client <ninguna> <ninguna> (no hay ninguna descripción disponible)
un dhcp3-client <ninguna> <ninguna> (no hay ninguna descripción disponible)
un dhcp3-common <ninguna> <ninguna> (no hay ninguna descripción disponible)
un dhcpd <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii isc-dhcp-client 4.2.4-7ubuntu12 i386 ISC DHCP client
ii isc-dhcp-common 4.2.4-7ubuntu12 i386 common files used by all the isc-dhcp* packages
un udhcp <ninguna> <ninguna> (no hay ninguna descripción disponible)
```

Vemos que solo está instalado el cliente (isc-dhcp-client), por lo que procederemos a instalar el paquete dhcp server (isc-dhcp-server o dhcp3-server depende de la versión de Ubuntu que tengamos):

```
#apt-get install isc-dhcp-server
```

```
alumno@alumno-VirtualBox ~ $ sudo apt-get install isc-dhcp-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  isc-dhcp-client isc-dhcp-common
Paquetes sugeridos:
  apparmor isc-dhcp-server-ldap
Se instalarán los siguientes paquetes NUEVOS:
  isc-dhcp-server
Se actualizarán los siguientes paquetes:
  isc-dhcp-client isc-dhcp-common
2 actualizados, 1 se instalarán, 0 para eliminar y 760 no actualizados.
Necesito descargar 2.066 kB de archivos.
Se utilizarán 2.279 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Mostramos nuevamente la lista de paquetes instalados para comprobar que ya aparece el dhcp-server:

```
#dpkg -l "*dhcp*"
```

```
alumno@alumno-VirtualBox ~ $ dpkg -l "*dhcp*"
Deseado=Desconocido/Instalar/Eliminar/Purgar/Retener
| Estado=No/Instalado/Config-files/Desempaquetado/Medio-conf/Medio-inst/espera-disparo/pendiente-disparo
|/ Err?=(ninguno)/Requiere-reinst (Estado,Err: mayúsc.=malo)
||/ Nombre Versión Arquitectura Descripción
+++-----+-----+-----+-----+
un dhcp <ninguna> <ninguna> (no hay ninguna descripción disponible)
un dhcp-client <ninguna> <ninguna> (no hay ninguna descripción disponible)
un dhcp3-client <ninguna> <ninguna> (no hay ninguna descripción disponible)
un dhcp3-common <ninguna> <ninguna> (no hay ninguna descripción disponible)
un dhcp3-server <ninguna> <ninguna> (no hay ninguna descripción disponible)
un dhcpd <ninguna> <ninguna> (no hay ninguna descripción disponible)
ii isc-dhcp-client 4.2.4-7ubuntu12.1 i386 ISC DHCP client
ii isc-dhcp-common 4.2.4-7ubuntu12.1 i386 common files used by all the isc-dhcp* packages
ii isc-dhcp-server 4.2.4-7ubuntu12.1 i386 ISC DHCP server for automatic IP address assignment
un isc-dhcp-server-ldap <ninguna> <ninguna> (no hay ninguna descripción disponible)
un udhcp <ninguna> <ninguna> (no hay ninguna descripción disponible)
```


Podemos ver información detallada del paquete instalado:

```
#dpkg -s isc-dhcp-server
```

```
alumno@alumno-VirtualBox ~ $ dpkg -s isc-dhcp-server
Package: isc-dhcp-server
Status: install ok installed
Priority: optional
Section: net
Installed-Size: 2224
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: i386
Source: isc-dhcp
Version: 4.2.4-7ubuntu12.13
Replaces: dhcp3-server
Provides: dhcp3-server
Depends: debconf (>= 0.5) | debconf-2.0, sysv-rc (>= 2.88dsf-24) | file-
2.8.2), isc-dhcp-common (= 4.2.4-7ubuntu12.13), lsb-base, adduser
Suggests: isc-dhcp-server-ldap, apparmor
```

También podemos usar las aplicaciones gráficas de instalación/desinstalación de software para facilitarnos el proceso de instalación. Estas aplicaciones son muy fáciles de usar y nos permiten instalar la mayoría del software disponible.

8.2 Inicio y parada del servicio DHCP.

Cuando hemos instalado el servicio DHCP, como se ha descrito anteriormente, el servicio queda preparado para iniciarse siempre durante el inicio del sistema. Pero, si queremos detener o iniciar el servicio en un momento dado, ¿qué tenemos que hacer? ¿Por qué puede ser necesario detener y reiniciar el servicio?

El **estado del servicio** puede ser en **ejecución** (running) o **parado** (stopped). Se puede consultar el estado con la opción **status** de la orden service:

La gestión del estado de un servicio normalmente incluye las opciones start, stop, status, restart y reload. Estas son las más usuales, pero cada servicio puede definir las que crea oportunas. Estos son ejemplos de gestión del estado del servicio DHCP:

- ✚ Se puede **arrancar el servicio** con la opción start del comando service:
- ✚ Se puede **detener el servicio** con la opción stop de la orden service:
- ✚ Se puede **iniciar de nuevo el servicio** (recargar) con la opción reload o restart del comando service:
- ✚ Para saber las opciones posibles de un servicio se puede hacer el truco:

```
usuario@usuario-VirtualBox ~ $ sudo service isc-dhcp-server patapum
Usage: /etc/init.d/isc-dhcp-server {start|stop|restart|force-reload|status}
```

Comprobar que el servidor DHCP está en funcionamiento es un proceso muy sencillo, basta comprobar que el servicio está encendido. Esto no quiere decir, en ningún caso, que el servicio esté funcionando correctamente. Quizás el servidor está encendido pero no está correctamente configurado. De hecho, la configuración es la parte realmente importante de la administración de un servicio, y también del servicio DHCP.



Aparte de comprobar el estado del servicio (con la opción **status**), el administrador puede asegurarse de que el demonio del servicio está en ejecución buscando su PID (Process Identifier o indicador de proceso). Otra actividad a hacer es monitorizar el registro de actividades del servicio (los logs). Todo el tráfico DHCP es tráfico de red TCP/IP; por lo tanto también se puede observar el estado de los puertos y analizar el tráfico que se produce.

Todo proceso en el sistema tiene un identificador de proceso. El PID de los servicios usualmente se guardan en el sistema de archivos (en el **directorio /var/run**) en forma de archivo que contiene un valor numérico (en texto) correspondiente al PID del proceso



Con el servicio en marcha siempre se puede observar el PID del servidor con algunas de las siguientes opciones:

```
usuario@usuario-VirtualBox /var/run $ ps -A | grep dhcp
1462 ?        00:00:00 dhcpd

usuario@usuario-VirtualBox /var/run $ service isc-dhcp-server status
isc-dhcp-server start/running, process 1462

usuario@usuario-VirtualBox /var/run $ ls -l /var/run/dhcp-server/dhcpd.pid
-rw-r--r-- 1 dhcpd dhcpd 5 sep 30 13:46 /var/run/dhcp-server/dhcpd.pid
usuario@usuario-VirtualBox /var/run $ cat /var/run/dhcp-server/dhcpd.pid
1462
```

Todos los servicios del sistema normalmente se monitorizan anotando en ficheros de texto un registro de todas las acciones que realizan, son los archivos conocidos como **archivos de log**. Tanto se puede utilizar un archivo genérico por el sistema como un archivo independiente para un servicio determinado. El servidor DHCP utiliza el archivo de monitorización estándar **/var/log/syslog**. En este archivo se registra cada vez que el servicio se pone en marcha y se detiene, entre otras cosas.

#cat /var/log/syslog | grep dhcp

```
Sep 30 14:03:24 usuario-VirtualBox dhcpd: DHCPDISCOVER from 08:00:27:fb:81:96 via eth0
Sep 30 14:03:24 usuario-VirtualBox dhcpd: ICMP Echo reply while lease 172.16.0.100 valid.
Sep 30 14:03:24 usuario-VirtualBox dhcpd: Abandoning IP address 172.16.0.100: pinged before offer
Sep 30 14:03:42 usuario-VirtualBox dhcpd: DHCPDISCOVER from 08:00:27:fb:81:96 via eth0
Sep 30 14:03:43 usuario-VirtualBox dhcpd: ns1.example.org: temporary name server failure
Sep 30 14:03:43 usuario-VirtualBox dhcpd: ns2.example.org: temporary name server failure
Sep 30 14:03:43 usuario-VirtualBox dhcpd: DHCPOFFER on 172.16.0.101 to 08:00:27:fb:81:96 (usuario-VirtualBox) via eth0
Sep 30 14:03:43 usuario-VirtualBox dhcpd: DHCPREQUEST for 172.16.0.101 (172.16.0.100) from 08:00:27:fb:81:96 (usuario-VirtualBox) via eth0
Sep 30 14:03:43 usuario-VirtualBox dhcpd: DHCPACK on 172.16.0.101 to 08:00:27:fb:81:96 (usuario-VirtualBox) via eth0
Sep 30 14:05:51 usuario-VirtualBox dhcpd: receive_packet failed on eth0: Network is down
```

El servicio DHCP guarda la información de **registro de las concesiones** que efectúa, en un fichero de **leases**. Esto le permite seguir la pista de las direcciones IP que ha concedido y mantener la consistencia entre varios arranques del mismo servidor. Se puede observar este archivo en **/var/lib/dhcp/dhcpd.leases**:

```
usuario@usuario-VirtualBox ~ $ cat /var/lib/dhcp/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.2.4

lease 172.16.0.100 {
    starts 3 2015/09/30 11:49:38;
    ends 3 2015/09/30 11:59:38;
    tstp 3 2015/09/30 11:59:38;
    cltt 3 2015/09/30 11:49:38;
    binding state free;
    hardware ethernet 08:00:27:fb:81:96;
}
server-uid "\000\001\000\001\035\236\210\010\000\373\201\226";

lease 172.16.0.100 {
    starts 3 2015/09/30 12:03:24;
    ends 3 2015/09/30 12:03:24;
    cltt 3 2015/09/30 12:03:24;
    binding state abandoned;
    next binding state free;
    rewind binding state free;
    client-hostname "usuario-VirtualBox";
}
```

Las tareas principales para configurar un servidor DHCP son las siguientes:

- ✚ Instalar el software del servidor DHCP.
- ✚ Activar / desactivar el servicio DHCP.
- ✚ Observar / hacer la lista de la configuración actual del servidor DHCP.
- ✚ Modificar la configuración del servidor DHCP.
- ✚ Monitorizar los *logs* del servicio DHCP y los archivos de registro de las concesiones (*leases*).

8.3 Archivos relacionados con la configuración y administración de DHCP

Tras haber instalado en Linux el servidor DHCP, en el sistema de archivos se han creado varios **archivos relacionados con la administración y configuración del servicio**. Es necesario que conozcas donde se encuentran esos archivos, cual es su nombre, que función tienen y como se pueden modificar o interpretar. En este apartado vamos a ver la información relativa al sistema Linux Ubuntu.

Los archivos y directorios más importantes relacionados con el servicio DHCP en Ubuntu son:

- ✚ **Directorio /etc/dhcp:** Contiene archivos de configuración relacionados con el servicio DHCP. El principal de los archivos es **dhcpd.conf**.
- ✚ **Archivo /etc/dhcp/dhcpd.conf** es el archivo de configuración del servidor. En el siguiente apartado describiremos cual es la sintaxis de este archivo.
- ✚ **Archivo /etc/dhcp/dhclient.conf** es un archivo de configuración del cliente DHCP. Se encuentra en cualquier ordenador con Ubuntu que tenga el cliente DHCP instalado. Su contenido indica cómo se comporta el cliente cuando solicita el servicio DHCP. Por ejemplo, en el archivo se indican cuales son los parámetros de red que solicita el cliente a los servidores DHCP.

- ✚ **Archivo `/etc/default/isc-dhcp-server`** establece los interfaces de red por los que el servidor DHCP atiende o escucha a los clientes.
- ✚ **Archivo `/var/lib/dhcp/dhcpd.leases`** contiene información actualizada sobre las concesiones que ha otorgado el servidor a los clientes. Dentro de este archivo hay una entrada por cada concesión que se ha dado y en la que se indica la IP que se ha concedido al cliente, la dirección física del cliente, cuanto tiempo de concesión se ha utilizado, etc.
- ✚ **Archivo `/usr/sbin/dhcpd`** es el archivo ejecutable correspondiente al servicio. Es un demonio que se encarga de escuchar las solicitudes de los clientes DHCP y controlar la entrega correcta de parámetros de red a los clientes.
- ✚ **Archivo `/var/log/syslog`** es un archivo de texto donde se registran los inicios y paradas de los servicios (registro de logs de servicios). Cuando se produzcan fallos al iniciar y detener el servicio DHCP debemos consultar la información que nos da este archivo sobre el fallo producido para intentar solucionarlo.

8.4 El archivo de configuración del servicio DHCP.

Cuando se inicia el servidor DHCP en Linux, se ejecuta el demonio correspondiente al servicio cuyo nombre es `dhcpd`. El archivo **`dhcpd.conf`** es un archivo de texto que es leído y establece la configuración correspondiente al servidor DHCP.

Siempre que hagamos modificaciones en este archivo, debemos reiniciar el servicio DHCP para que trabaje con la nueva configuración.



Si se producen fallos al iniciar el servicio DHCP, se guarda información sobre el fallo en el archivo registro de logs de servicio. En ese archivo se indica el tipo de fallo y la parte del archivo de configuración que ha producido el fallo.

El archivo consta de una secuencia de **sentencias** o directivas de dos tipos:

- ✚ **Parámetros.**
- ✚ **Declaraciones.**

Los **parámetros** permiten establecer una opción de configuración del servicio. En los parámetros se puede asignar un valor o un conjunto de valores, que determinan una condición de funcionamiento del servidor o el valor de parámetro que se entrega a los clientes. Se usan principalmente dos sintaxis para asignar valores a los parámetros:

- ✚ `Nombre_parámetro;`
- ✚ `Nombre de parámetro valores;`

La primera sintaxis significa que está activado el parámetro. La segunda sintaxis permite asignar uno o varios valores a los parámetros. **Si se asignan varios valores, estos se separan con espacios.**

Dentro de las declaraciones se pueden incluir parámetros e incluso otras declaraciones. Las declaraciones tienen la sintaxis





```
Declaración {  
    [parámetros]  
    [declaraciones]  
}
```

8.4.1 Declaraciones

Vamos a ver los **tipos de declaraciones** que se pueden establecer en el archivo de configuración **`dhcpd.conf`** del servidor DHCP en los sistemas Linux y que nos van a permitir declarar la red en la que trabaja el servidor y el conjunto de direcciones IP que concede el servidor en la red.

Las declaraciones que podemos utilizar son:

- ✚ **Subnet.** `subnet subnet-number netmask netmask {[parameters] [Declarations]}`
Permite definir opciones para una subred concreta. Es la sentencia más usual en las definiciones de configuración del servidor DHCP.
Dentro de esa declaración es obligatorio especificar al menos un conjunto de direcciones que otorga el servidor en la red. A este conjunto de direcciones se le denomina rango.

-  **Host.** `host hostname {[parameters] [Declarations]}`
Proporciona un ámbito de definición para un equipo concreto. Las opciones que se definen dentro de una sentencia `host` afectan únicamente al equipo indicado. Se requiere una sentencia `host` para poder hacer asignaciones dinámicas fijas (asignar siempre la misma IP basándose en la MAC).
-  **Shared-network:** `shared-network name {[parameters] [Declarations] }`
Permite agrupar varias subredes (`subnet`) en una misma declaración.
Se utiliza cuando una misma red física se compone de varias subredes lógicas.
-  **Group.** `group {[parameters] [Declarations]}`
Se utiliza para agrupar declaraciones de manera que las opciones definidas afecten al grupo de elementos que contiene. Estos pueden ser `shared-networks`, `Subnet`, `hosts` y hasta otros grupos.
-  **Range:** `range [dynamic-bootp] low-address [high-address];`
Indica el intervalo de direcciones dinámicas disponibles para asignar. El servidor DHCP extrae las direcciones dinámicas de este intervalo de direcciones.

Ejemplo de sintaxis:

```
subnet IP_red netmask mascara_de_red {  
    range IP_menor IP_mayor;  
    [parámetros]  
}
```

Es adecuado hacer reservas para ordenadores de la red que vayan a ser servidores dentro de la red. En una declaración `host` se da un nombre a la declaración que corresponderá al nombre del equipo para el que se hace la reserva y se indica la dirección física o MAC del equipo y la IP que se le va a asignar. La sintaxis de esta declaración es:

```
host nombre {  
    [parámetros]  
    hardware ethertnet direccion_MAC;  
    fixed-address dirección_IP;  
}
```

Las declaraciones **shared-network** y **group** permiten simplificar el archivo de configuración agrupando en una declaración varias declaraciones. Los parámetros que se asignen dentro de una declaración `shared-network` o `group` funcionarán como parámetros globales para todas las declaraciones que se hagan dentro.

La declaración **shared-network** nos permite describir una red que está dividida en varias subredes. Por tanto, dentro de una declaración `shared-network` tendremos varias declaraciones `subnet`. El uso de `shared-network` es problemático ya que no asegura que unos equipos reciban o no IP dentro de una subred concreta. Los parámetros que se establezcan dentro de la declaración `shared-network` configurarán todas las subredes de la `shared network`. La sintaxis es:

```
shared-network nombre {  
    [parámetros y declaraciones]  
    Declaración subnet de primera subred  
    [otras declaraciones subnet]  
}
```



La declaración **group** permite incluir varias declaraciones `host` y se usa para aplicar los mismos parámetros a todas las declaraciones que se realicen dentro de ella. Se pueden usar varias declaraciones `group` en el archivo de configuración. La sintaxis de una declaración `group` es:

```
group nombre {  
    [parámetros y declaraciones host]  
}
```

8.4.2 Parámetros

Al iniciar la unidad hablábamos de que un servidor de asignación automática de parámetros de configuración red asignaba IP a los clientes y otros muchos parámetros como puerta de enlace, nombre del dominio DNS, servidores DNS etc. Ahora aprenderemos a establecer en los sistemas Linux los parámetros de red que se entregarán a los clientes así como opciones de configuración del funcionamiento del servidor DHCP.

Recordemos que en el archivo **dhcpd.conf** hay dos tipos de sentencias o directivas: parámetros y declaraciones. Los parámetros permiten establecer los “parámetros” de red que asigna el servidor DHCP a los clientes y opciones de funcionamiento del servicio. Un mismo parámetro se puede declarar en varias partes del archivo de configuración, pudiéndole asignar distintos valores. Dependiendo de donde se declara un parámetro podemos hablar de:

-  **Parámetros Globales:** Se declaran fuera de las sentencias de declaración y afectan a todos los clientes del servicio.
-  **Parámetros Locales:** Se declaran dentro de una sentencia de declaración y afectan sólo a los clientes definidos en esa declaración. Si a un parámetro local se le ha asignado un valor de forma global, en el ámbito local prevalece el valor asignado de forma local.

Todos los parámetros que se declaren con **option** son parámetros que el servidor puede entregar a los clientes. En la siguiente tabla se describe la sintaxis de los principales parámetros que podemos encontrar en un archivo de configuración. En la sintaxis, se representan en cursiva, los valores que se asignan a cada parámetro.

Sintaxis.	Descripción.
authoritative;	Implica que el servidor es autoritativo en la red. El servidor reasignará IP a los clientes que detecte mal configurados. La sentencia opuesta a ésta es " not authoritative ".
default-lease-time <i>segundos</i> ;	Tiempo de concesión que se otorgará a los clientes por defecto, es decir, cuando éstos no hayan solicitado otro.
max-lease-time <i>segundos</i> ;	El máximo tiempo de concesión que se puede otorgar a los clientes.
min-lease-time <i>segundos</i> ;	El mínimo tiempo de concesión que se puede otorgar a los clientes.
range <i>ip_menor ip_mayor</i> ;	Un rango o conjunto de IP que otorgará el servidor a los clientes. Dará IP comprendidas entre la IP menor y la IP mayor, ambas incluidas. Este parámetro debe estar incluido dentro de una declaración subnet .
hardware <i>tipo dirección_física</i> ;	Permite indicar cuál es la dirección física de un cliente DHCP. En tipo se indica el tipo de adaptador de red del cliente (normalmente ethernet).
fixed-address <i>IP</i> ;	Permite indicar cuál es la IP que se reserva para un cliente concreto. Este parámetro se debe incluir en una declaración host y está asociado al parámetro fixed-address.
option subnet-mask <i>máscara</i>	Indica la máscara de red que usarán los clientes.
option broadcast-address <i>dirección</i> ;	Indica cual es la dirección IP de broadcast que usarán los clientes.
option routers <i>IP</i> ;	Indica cual es la dirección IP de la puerta de enlace que se entregará a los clientes.
option domain-name <i>"nom_dominio"</i> ;	Indica cual es el nombre de dominio que usará el cliente como dominio de pertenencia.
option domain-name-servers <i>servidores</i> ;	Indica cuales son los servidores DNS que deben usar los clientes. Generalmente se indican las IP de éstos y se separan con coma.

Ejemplo de configuración del DHCP básica:

```
# Opciones globales del servidor DHCP (usuales)
ddns-update-style none;

# Definición de la red a la que se ofrece el servicio DHCP
subnet 192.168.0.0 netmask 255.255.255.0 {
    # Opciones genéricas para todos los equipos de la red
```

```
option routers 192.168.0.1;
option subnet-mask 255.255.255.0;
option domain-name "iessidon.org";
option domain-name-servers 192.168.1.1;
# Definición del intervalo de IPs dinámicas a usar
# Y los tiempos de las concesiones
range 192.168.0.128 192.168.0.254;
default-lease-time 21600;
max-lease-time 43200;

# Opciones de equipos individuales
# El servidor imp obtiene siempre una dirección fija basada en MAC
host imp {
    hardware ethernet 12: 34: 56: 78: AB: CD;
    fixed-address 207.175.42.254;
}
}
```

Ejemplo de configuración del DHCP avanzada

```
ddns-update-style interim;

subnet 192.168.1.0 netmask 255.255.255.0 {
    #---default gateway

    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
    option nis-domain "domain.org";
    option domain-name "domain.org";
    option domain-name-servers 192.168.1.1;
    default-lease-time 21600;
    max-lease-time 43200;

    # We want the nameserver to appear at a fixed address
    host ns {
        hardware ethernet 12: 34: 56: 78: AB: CD;
        fixed-address 207.175.42.254;
        default-lease-time 43200;
        max-lease-time 86400;
    }
}
```

Ejemplo de configuración del DHCP avanzada II

```
subnet 10.0.0.0 netmask 255.255.255.0 {
    option routers 10.0.0.254;

    group {
        filename "osProfesores";
        next-server ncd-booter;
        host ncd2 {hardware ethernet 0:c0:c3:88:2d:81; }
        host ncd3 {hardware ethernet 0:c0:c3:00:14:11; }
    }
    # Unknown clientes get this pool.
    pool {
        option domain-name-servers sidon.com;
        max-lease-time 300;
    }
}
```



```
        range 10.0.0.200 10.0.0.253;
        allow unknown-clientes;
    }

    # Known clientes get this pool.
    pool {
        option domain-name-servers ns1.example.com, ns2.example.com;
        max-lease-time 28800;
        range 10.0.0.5 10.0.0.199;
        deny unknown-clientes;
    }
}
```

9 El Servicio DNS

Todos los nodos de una red deben tener una **dirección IP**. El protocolo IP utiliza esta dirección para identificar los distintos nodos de Internet. Las direcciones IP se organizan en cuatro grupos de 8 bits y una dirección se representa de esta forma: 208.86.217.103.

Las direcciones IP son difíciles de recordar y, por lo tanto, es muy engorroso su manejo. A las personas nos resulta más fácil acordarnos de `www.google.es` que de un conjunto de números.

El **DNS** (siglas en inglés de **Domain Name System**, traducido significa Sistema de Nombres de Dominio) tiene por objeto facilitar esta tarea y permite traducir una dirección del tipo `www.google.es` a una dirección IP y viceversa.

Por lo tanto, DNS es una base de datos distribuida, con información que se usa para traducir los nombres de dominio en números de protocolo de Internet (IP).



Los nombres son más fáciles de recordar y usar por las personas, pero hay que tener en cuenta, que la expresión numérica es la forma en que las máquinas pueden encontrarse en Internet.

9.1 Sistemas de nombres planos y jerárquicos.

A principios de la década de los 80, las páginas Web no existían y el número de servidores era escaso. Entonces todos los Host tenían la lista de nombres de dominio y la actualizaban diariamente. Se trataba de un **sistema de nombres planos**.

En el archivo **hosts.txt** en Windows o **hosts** en Linux, se tenía una lista de nombres con su respectiva dirección IP. Todavía se pueden usar estos ficheros a nivel de una LAN.

Hoy en día es imposible que puedas manejar todos los nombres nuevos que se crean a diario y, por este motivo, se utiliza una forma de gestión jerárquica. El DNS es el servicio

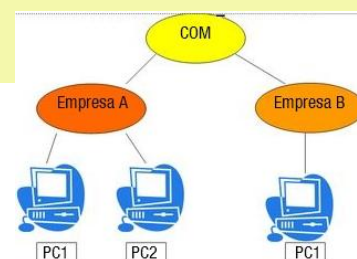
encargado de esta gestión consultando una base de datos distribuida, formada por todos los servidores DNS. A este sistema se le denomina **sistema de nombres de dominio jerárquico**.

Los sistemas de nombres de dominio jerárquico son aquellos en los que existe una jerarquía a la hora de construir el nombre completo de este ordenador. De esta forma, podemos determinar su ubicación geográfica o el departamento al que pertenece dentro de una empresa. Empleando este tipo de sistema puedes poner el nombre que quieras a un PC, la única limitación es que no pueden estar en la misma empresa.

Copyright (c) 1993-1999 Microsoft Corp.

```
#
# Éste es un ejemplo de archivo HOSTS usado por Microsoft TCP/IP para Windows.
#
# Este archivo contiene las asignaciones de las direcciones IP a los nombres de
# host. Cada entrada debe permanecer en una línea individual. La dirección IP
# debe ponerse en la primera columna, seguida del nombre de host correspondiente.
# La dirección IP y el nombre de host deben separarse con al menos un espacio.
#
# También pueden insertarse comentarios (como éste) en líneas individuales
# o a continuación del nombre de equipo indicándolos con el símbolo ";"
#
# Por ejemplo:
# 102.54.94.97 rhino.acme.com # servidor origen
# 38.25.63.10 x.acme.com # host cliente x
```

```
127.0.0.1 localhost
192.168.0.2 mi-servidor.lan
192.168.0.3 otro-ordenador.lan
```



Un sistema de nombres plano no se puede usar en redes grandes como Internet, dado que no se puede repetir el nombre de dos ordenadores en la red y sería difícil para los administradores buscar nombres que no se repitan. Por otra parte, la gestión del sistema de nombres planos debería estar totalmente centralizada. Con un sistema de nombres jerárquico como el usado en Internet, se pueden tener muchos ordenadores con nombre `www` siempre que pertenezcan a diferentes organizaciones o dominios y la **gestión de los nombres está distribuida**.



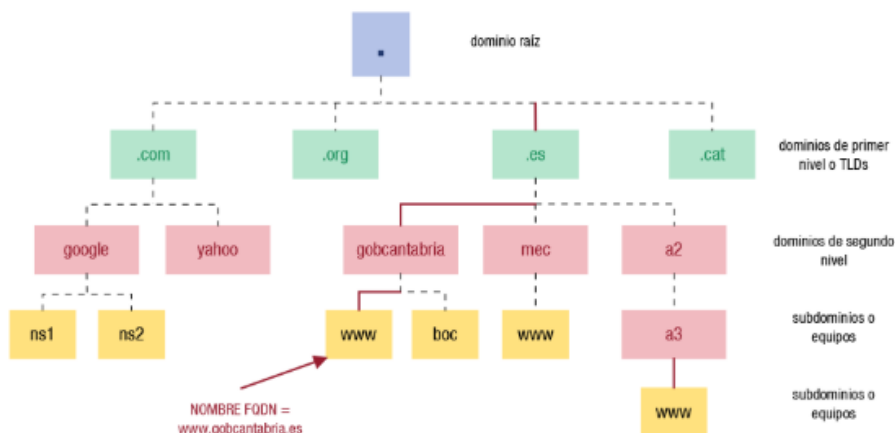
Los organismos que alquilan los nombres de dominio son:

- ✚ A nivel internacional, Internet Corporation for Assigned Names and Numbers (**ICANN**), (traducido significa Corporación de Internet para la Asignación de Nombres y Números).
- ✚ En el caso de los `.es`, los reparte **nic.es**, que forma parte de red.es del Gobierno de España.

9.2 Espacio de nombres de dominio.

El espacio de nombres de dominio es una **base de datos distribuida** entre múltiples servidores DNS y que almacena los nombres DNS de equipos junto con sus direcciones IP. Es una **estructura jerárquica** organizada en forma de árbol con varios niveles de dominio.

La siguiente imagen muestra una pequeña parte de la estructura jerárquica del espacio de nombres DNS:



Cada elemento del árbol se etiqueta con un nombre de hasta 63 caracteres. Y cada elemento se identifica en el sistema de nombres con un FQDN (siglas en inglés de, **Fully Qualified Domain Name**, traducido significa nombre de dominio completo). Un **FQDN** representa el nombre de un elemento cualquiera en el espacio de nombres. Un nombre FQDN tiene esta estructura:

nombreEquipo.subdominio.dominiosegundonivel.dominioprimernivel.

Obviamente si un equipo no pertenece a un subdominio, sino que pertenece a un dominio de segundo nivel, el nombre de subdominio no se usará en el FQDN. Si el equipo estuviera dentro de varios niveles de subdominios, estos se especificarían en el FQDN separados por puntos y leídos desde el inferior al superior. Un FQDN puede contener hasta un máximo de 255 caracteres y no puede contener algunos caracteres como, por ejemplo, la letra "ñ".

Un FQDN termina en un punto que representa al dominio raíz. Esto es importante, especialmente cuando estemos realizando la configuración de un servidor DNS:



Ejercicio: Obtén los FQDN de cada uno de los ordenadores de la imagen de arriba que representa la estructura jerárquica del espacio de nombres DNS.

Sol: ns1.google.com. , ns2.google.com. , www.gobicantabria.es. , boc.gobicantabria.es. , www.mec.es. , www.a3.a2.es.

9.3 Tipos de dominio

Antes hemos estudiado el espacio de nombres DNS y hemos visto que está basado en una estructura jerárquica basada en distintos niveles de dominio. Respecto de esa estructura jerárquica, existen los siguientes **tipos de dominios**:

- ✚ **Dominio raíz:** de este dominio cuelga toda la estructura del espacio de nombres DNS. Se simboliza con un punto. Bajo el directorio raíz hay dominios de primer nivel. Se encarga de gestionar la información sobre los dominios de primer nivel, en concreto, sobre los nombres de esos dominios y sobre los servidores encargados de la gestión de esos dominios. El organismo que se encarga de su gestión es **ICANN**.
- ✚ **Dominios de primer nivel (TLD – Top Level Domain):** son dominios que en la estructura del espacio de nombres DNS se encuentran bajo el dominio raíz. Dentro de este tipo de dominio podemos realizar la siguiente subclasificación:
 - **Dominios genéricos o gTLD** (sigla en inglés de Generic Top Level Domain) son aquellos que tienen tres o más letras: .com, .org, .info, .pro son algunos ejemplos. Inicialmente pensados para una clase particular de organizaciones (por ejemplo, .com para organizaciones comerciales), actualmente la mayoría de ellos pueden usarse sin restricción. No obstante se mantienen una serie de ellos para usarse de manera restringida.

- **Dominios geográficos o ccTLD** (sigla en inglés de Country-Code Top_Level Domain) son los formados por dos letras y en general hacen referencia a un país (Asignados por la ISO 3166-1): .es, .uk, .us, .fr son algunos ejemplos.
 - **Dominio .arpa**: es una excepción y por eso aparece aparte. **in-addr.arpa** es usado por los servidores DNS inversos para la obtención del FQDN de una dirección IP (búsqueda DNS inversa). Por ejemplo la dirección 212.30.222.56 es mapeada al nombre 56.222.30.212.in-addr.arpa.
- ✚ **Dominios de segundo nivel**: son los dominios que se encuentran bajo los TLDs. Cada uno de estos dominios está registrado a favor de una determinada entidad (empresa, universidad, órgano, persona, etc.). La entidad propietaria del dominio es la encargada de la gestión del dominio. Para un dominio de este tipo se tienen uno o varios servidores DNS que tienen información sobre máquinas disponibles en el dominio, sobre posibles subdominios y sobre servidores DNS del dominio y de los subdominios. Cuando una entidad desea disponer de un dominio, debe registrarlo ante un registrador oficial autorizado por ICANN. Dominios de segundo nivel son wikipedia.org, mec.es, google.com y otros muchos.
- ✚ **Subdominios**: Son dominios que hay bajo un dominio de segundo nivel o bajo otro subdominio. Un subdominio no tiene que ser registrado como un dominio de segundo nivel. Es el propietario del dominio de segundo nivel quien decide la existencia o no de subdominios. En un subdominio puede haber servidores encargados de toda la gestión del subdominio aunque también esa gestión se puede llevar a cabo desde los servidores de segundo nivel. Para cada subdominio se tiene información sobre las máquinas y servidores pertenecientes al subdominio.

Whois es una base de datos distribuida que te puede informar de los datos de un dominio DNS. Esta base de datos puede ser usada para obtener información sobre el usuario registrante, el registrador o el listado de servidores de nombres.



9.4 Delegación DNS

La delegación DNS es el proceso por el cual el gestor de un determinado dominio delega la gestión del mismo a otra entidad. Por ejemplo, ICANN delega la gestión del ccTLD en la empresa pública Red.es, esto significa que ICANN ya no se encargará de gestionar este dominio.

Este proceso se puede repetir varias veces como de hecho suele ocurrir. Red.es deberá gestionar el alta de los dominios como eldigital.es. Cuando Red.es crea este dominio automáticamente lo delega a la empresa El Asidonense SA para que, de esta forma, sea la propia empresa la que dé de alta, por ejemplo, www.eldigital.es y lo más importante para que sea la empresa la que gestione los servidores DNS.

Alquilar un nombre de dominio es fácil. Suele resultar más económico si lo alquilas con los servicios de host (espacio para hospedaje de páginas Web y correo electrónico).



9.5 Funcionamiento del servicio DNS.

Después de todo este lío con los nombres de dominio vamos a ver cómo funciona el servicio de DNS. Se basa en una consulta del cliente a un servidor DNS. El sistema operativo suele tener configurados por el usuario, al menos, dos servidores DNS uno primario y otro secundario. Pero si se tiene configurada la opción DHCP, se configuran automáticamente como viste en la unidad anterior. La consulta se envía al servidor DNS primario y si éste no contesta se usa el secundario. La comunicación se realiza por el **puerto 53**.

Los clientes DNS realizan sus consultas a través de resolutores. Un **resolutor** es un proceso que se ejecuta a petición de un programa que usa un nombre DNS para establecer una conexión. El resolutor gestiona el proceso de consulta del nombre DNS, recepción de la respuesta relativa a la consulta y entrega del resultado. Los procesos resolutores se ejecutan en los clientes DNS y en servidores que deben dar respuestas a clientes DNS.

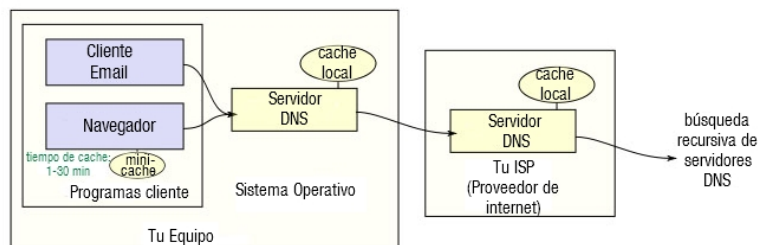


Cualquier aplicación que ejecutemos en nuestro ordenador y que establezca conexiones con otros ordenadores a partir de sus nombres DNS, lanzará a ejecución un cliente DNS para gestionar la resolución de un nombre DNS.

El resolutor, realizará su función de la siguiente forma:

1. El resolutor consulta el nombre del dominio en una caché local DNS donde se almacenan los datos sobre los últimos nombres DNS consultados. Si se encuentra en la caché, devuelve la IP del nombre DNS al cliente que realizó la petición y termina el proceso.

- Si no se encuentra el nombre DNS en la caché, lo busca en un archivo local de nombres (en Linux /etc/hosts). Si lo encuentra, devuelve la IP del nombre DNS al cliente y termina el proceso.
- Si no se encontró el nombre de dominio, el resolutor establece, como cliente, una conexión con el primer servidor DNS y le envía un mensaje UDP al puerto 53 consultándole el nombre de dominio en cuestión.
- El servidor DNS consulta primero su caché DNS para resolver el nombre y después comprueba si puede resolver el nombre en un archivo de registros de zona (base de datos de nombres) que contiene nombres que está autorizado para resolver el servidor.
- En el caso de no encontrarse, el servidor inicia un proceso de consulta que se explicará en el siguiente apartado con más detalle a otros servidores DNS del sistema DNS. Finalmente el servidor DNS del dominio buscado, envía al primer servidor DNS la dirección IP del ordenador perteneciente a ese dominio.
- El primer servidor DNS enviará al resolutor la dirección IP del nombre de dominio buscado y el resolutor se la entregará al cliente que le solicitó la consulta.

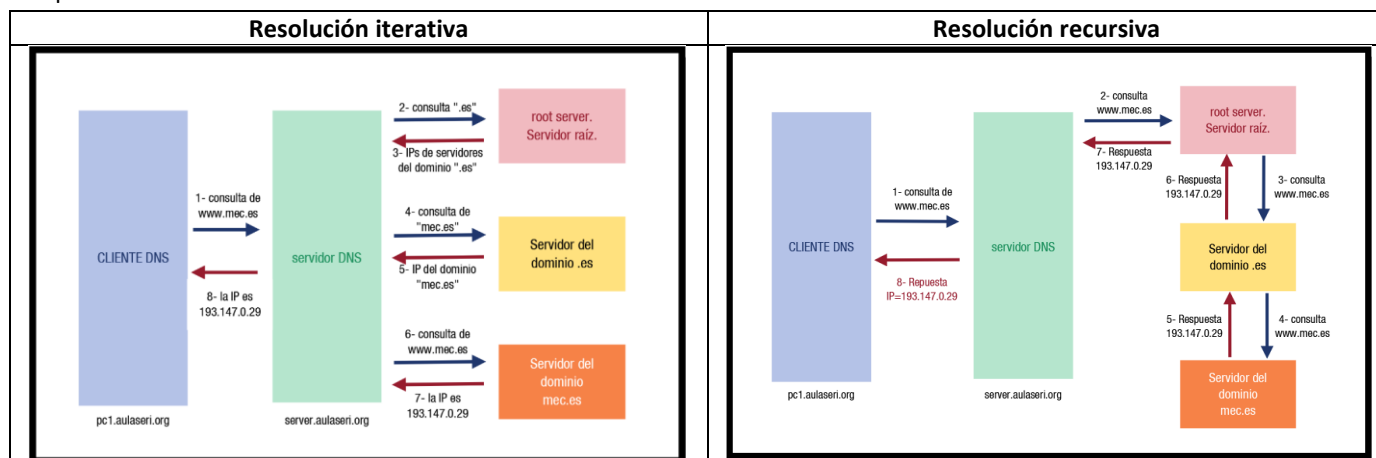


9.5.1 Proceso de resolución de un nombre de dominio

Las resoluciones de nombres de dominio en el sistema de dominios pueden ser de dos tipos:

- Resolución recursiva.
- Resolución iterativa.

Vamos a suponer que un cliente DNS en el equipo pc1.aulasri.org del dominio local aulasri.org solicita la resolución del nombre www.mec.es. El cliente DNS tiene configurado como primer servidor DNS el equipo server.aulasri.org perteneciente a nuestro dominio.



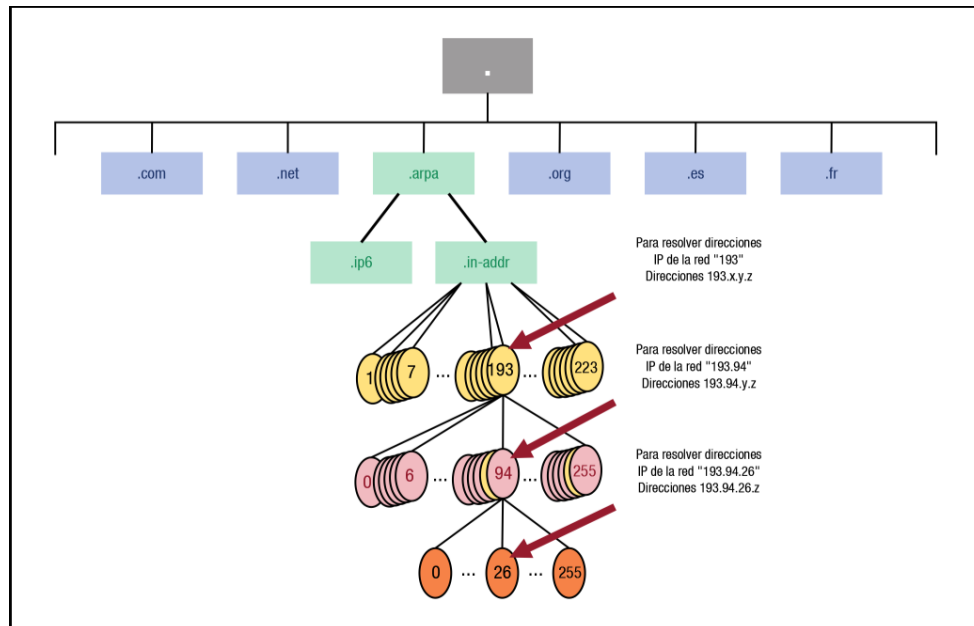
9.5.2 Resoluciones directas y resoluciones inversas.

Las resoluciones y consultas DNS pueden ser de dos tipos:

- Directas:** Cuando se trata de obtener la dirección IP que corresponde a un nombre DNS.
- Inversas:** Cuando se trata de obtener el nombre DNS correspondiente a una dirección IP.

Para las **resoluciones inversas** se utiliza el dominio de primer nivel **".arpa"** con sus correspondientes servidores. Dentro de ese dominio se tiene el dominio **"in-addr.arpa"** para resolver direcciones IPv4 y el dominio **"ip6.arpa"** para resolver direcciones IPv6.

Cualquier organización que tenga en propiedad una dirección de red debe responsabilizarse de tener servidores DNS que resuelvan inversamente las direcciones IP pertenecientes a la dirección de red. La siguiente imagen representa como están organizados los servidores para las resoluciones inversas y se indica, para algunos de ellos, las direcciones IP en las que participan en su resolución.



Cuando un cliente DNS realiza una consulta inversa de una dirección IPv4 debe preguntar por un nombre formado por la dirección IP escrita al revés y seguida de un nombre del dominio ".in-addr.arpa" y finalizado en un punto que especifica el servidor raíz.

10 Base de Datos

10.1 Estructura

En este apartado vamos a conocer la estructura de estos registros. Cada servidor de nombres de dominio mantiene:

- ✚ Una base de datos que sirve para asociar los nombres de dominios con direcciones IP. Esta base de datos se conoce con el nombre de **archivos de la zona**.
- ✚ Cada servidor de nombres de dominio también mantiene una base de datos de resolución inversa. Esta base de datos se conoce con el nombre de archivos de **resolución inversa de la zona**.

Ambas bases de datos son manejadas por un servidor de nombres, el cual responde a las solicitudes hechas por el resolutor (resolver). El formato de dichas bases de datos son archivos de texto donde se definen los **registros de recurso "Resource Records, RR"**, que sirven para especificar la relación entre un nombre de dominio y una dirección IP. Además, sirve para especificar a qué zona del espacio de nombres de dominios, pertenece el servidor de nombres de dominios.

Para resolver nombres, los servidores consultan sus zonas. Las zonas contienen registros de recursos que constituyen la información de recursos asociada al dominio DNS. Por ejemplo, ciertos registros de recursos asignan nombres descriptivos a direcciones IP.

El formato de cada registro de recursos es el siguiente: **Propietario TTL Clase Tipo RDATA**

Donde:

- ✚ **Propietario:** es el nombre de host o del dominio DNS al que pertenece este recurso. Puede contener:
 - Un nombre de host/dominio completamente cualificado o no.
 - El símbolo "@" que representa el nombre de la zona que se está describiendo.
 - Una cadena vacía, en cuyo caso, equivale al propietario del registro de recursos anterior.

- ✚ **TTL: (Time To Live)** Tiempo de vida, indica el tiempo de vida durante el cual esa entrada puede ser considerada válida, es decir, el tiempo durante el cual se almacena esta entrada en la caché. Este campo es opcional. También se puede expresar mediante letras indicando días (d), horas (h), minutos (m) y segundos (s). Por ejemplo: "2h30m".
- ✚ **Clase:** define la familia de protocolos en uso. Suele ser siempre "IN", que representa Internet.
- ✚ **Tipo:** identifica el tipo de registro.
- ✚ **RDATA:** los datos del registro de recursos.

BIND (siglas en inglés de Berkeley Internet Name Domain, traducido significa Berkeley Internet nombre de dominio) es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix, en los cuales es un Estándar de facto.

En BIND el archivo de configuración se llama **named.conf** y los registros están en otros archivos que suelen denominarse **db**.

A continuación se describen los principales tipos de registros de recursos:

- ✚ **SOA**, siglas en inglés de Start Of Authority (Inicio de autoridad). Este registro especifica información de los DNS
- ✚ **NS**, siglas en inglés de Name Server (Nombre de servidor). Indica los servidores de DNS autorizados (principales y secundarios) para el dominio.
- ✚ **A**, siglas en inglés de Address (Dirección). Permite enlazar un nombre de dominio o subdominio hacia una dirección IPv4
- ✚ **Registro AAAA y A6**. Son muy similares al registro A, pero en lugar de apuntar a una dirección IPv4 apunta a una IPv6. No obstante, el registro A6 aún está en fase experimental por lo que de momento se recomienda el uso de AAAA:
- ✚ **PTR**, siglas en inglés de PoinTeR (Puntero). Son usados principalmente para la resolución inversa de nombres.
- ✚ **CNAME**, siglas en inglés de Canonical NAME (Nombre canónico). También se conoce como **Registro Alias**, permite apuntar un dominio hacia un nombre de servidor (host) y, por tanto, éste a la IP a la que está asociado el host.
- ✚ **MX**, siglas en inglés de Mail Exchange (Intercambio de correo). Especifica el servidor de email responsable de distribuir los emails para tu dominio
- ✚ **SRV**, siglas en inglés de SeRVice (Servicio). Suele ser utilizado para la configuración de servicios como Office 365 de Microsoft y para algunos protocolos XMPP, SIP o LDAP.

10.1.1 Tipos de registros SOA.

Aquí vas a ver como es el registro SOA. En él, entre otras cosas, tienes que indicar la dirección del servidor principal. Puedes entender que es uno de los más importantes.

Registro de Recurso SOA:

Cada zona contiene un registro de recursos denominado Inicio de Autoridad o SOA al comienzo de la zona. Los registros SOA incluyen los siguientes campos (sólo se incluyen los que poseen un significado específico para el tipo de registro):

- ✚ **Propietario:** nombre de dominio de la zona.
- ✚ **Tipo:** "SOA".
- ✚ **Persona responsable:** contiene la dirección de correo electrónico del responsable de la zona. En esta dirección de correo se utiliza un punto en el lugar del símbolo "@".
- ✚ **Número de serie:** muestra el número de versión de la zona, es decir, un número que sirve de referencia a los servidores secundarios de la zona para saber cuándo deben proceder a una actualización de su base de datos de la zona (o transferencia de zona). Cuando el número de serie del servidor secundario sea menor que el número del maestro, esto significa que el maestro ha cambiado la zona, y por tanto el secundario debe solicitar al maestro una transferencia de zona. Por tanto, este número debe ser incrementado (manualmente) por el administrador de la zona cada vez que realiza un cambio en algún registro de la zona (en el servidor maestro).

- ✚ **Actualización:** muestra cada cuánto tiempo un servidor secundario debe ponerse en contacto con el maestro para comprobar si ha habido cambios en la zona, y por tanto pedir una **transferencia de zona**.
- ✚ **Reintentos:** Tiempo que espera un servidor de nombres secundario para iniciar una nueva transferencia de zona en el caso de que falle este procedimiento.
- ✚ **Caducidad:** define el tiempo que el servidor secundario de la zona, después de la transferencia de zona anterior, responderá a las consultas de la zona antes de descartar la suya propia como no válida.
- ✚ **TTL (Time To Live):** este campo especifica el tiempo de validez (o de vida). Indica el intervalo de tiempo que el servidor esclavo estará haciendo reintentos de conexión con el servidor maestro porque este no contesta. Hasta que no pase este tiempo, aunque el servidor esclavo no se pueda actualizar, responde a las consultas de forma autoritaria, pero llegado el momento de expiración, deja de contestar a las consultas de la zona que no ha podido actualizarse.

Esta directiva se especifica así:

\$TTL tiempo

Por ejemplo, un tiempo de vida por defecto de 30 minutos se establecería así: **\$TTL 30m**

Un ejemplo de registro SOA sería el siguiente:

```
admon.com. IN SOA pc0100.admon.com hostmaster.admon.com. (
    2019102600 ; Serial Number (número de serie)
    1d12h      ; Refresh (Actualización) - 1día + 12 horas
    15m        ; Retry (Reintento) - 15 minutos
    3w12h      ; Expiry (Caducidad) - 3 semanas + 12 horas
    2h20m      ; TTL 2horas + 20 min
)
```

10.1.2 Tipos de registros NS

En el **registro NS** tenemos que indicar el FQDN de los servidores de dominio. Deben existir tantos registros NS como servidores de nombres tengas para la zona.

Registro de Recurso NS:

El registro de recursos NS indica los servidores de nombres autorizados para la zona. Cada zona debe contener registros indicando tanto los servidores principales como los secundarios. Por tanto, cada zona debe contener, como mínimo, un registro NS.

Por otra parte, estos registros también se utilizan para indicar quiénes son los servidores de nombres con autoridad en subdominios delegados, por lo que la zona contendrá, al menos, un registro NS por cada subdominio que haya delegado.

Ejemplos de registros NS serían los siguientes:

```
admon.com. IN NS pc0100.admon.com.
cadiz.admon.com. IN NS pc0102.cadiz.admon.com.
```

Esta lista de servidores de dominio es lo que necesita cualquier servidor DNS para obtener los datos de la zona.

10.1.3 Tipos de registros A, PTR, CNAME Y MX.

Los registros que hemos visto anteriormente son importantes pero no te permiten especificar una dirección IP para un determinado nombre. Esto es lo que hace un **registro A**.

Registro de Recurso A

El tipo de registro de recursos A asigna un nombre de dominio completamente cualificado (FQDN) a una dirección IP, para que los clientes puedan solicitar la dirección IP de un nombre de host dado. Un ejemplo de registro A que asignaría la dirección IP 158.42.178.1 al nombre de dominio sería pc0101.cadiz.admon.com, sería el siguiente:

```
pc0101.cadiz.admon.com. IN A 158.42.178.1
```

Registro de Recurso PTR

El registro de recursos PTR o puntero, realiza la acción contraria al registro de tipo A, es decir, asigna un nombre de dominio completamente cualificado a una dirección IP. Este tipo de recursos se utilizan en la denominada resolución inversa. Un ejemplo de registro PTR que asignaría el nombre pc0101.cadiz.admon.com a la dirección IP 158.42.178.1 sería:

```
1.178.42.158.in-addr.arpa. IN PTR pc0101.admon.cadiz.com.
```

Registro de Recurso CNAME

El registro de nombre canónico CNAME crea un alias (un sinónimo) para el nombre de dominio especificado. Un ejemplo de registro CNAME que asignaría el alias controlador al nombre de dominio pc0101.cadiz.admon.com, sería el siguiente:

```
controlador.valencia.admon.com. IN CNAME pc0101.valencia.admon.com.
```

Registro de Recurso MX

El registro de recurso de intercambio de correo (MX) especifica un servidor de intercambio de correo para un nombre de dominio. Puesto que un mismo dominio puede contener diferentes servidores de correo, el registro MX puede indicar un valor numérico que permite especificar el orden en que los clientes deben intentar contactar con dichos servidores de correo.

Un ejemplo de registro de recurso MX que define al servidor pc0100 como el servidor de correo del dominio admon.com, sería el siguiente:

```
admon.com. IN MX 0 pc0100.admon.com.
```

10.1.4 Tipo de registro SRV.

Por último, vamos a ver el registro SRV, aunque existen 30 tipos de registros, hasta aquí hemos visto los más importantes.

Con registros MX se puede especificar varios servidores de correo en un dominio DNS. De esta forma, cuando un proveedor de servicio de envío de correo necesite enviar correo electrónico a un host en el dominio, podrá encontrar la ubicación de un servidor de intercambio de correo. Sin embargo, ésta no es la forma de resolver los servidores que proporcionan otros servicios de red como WWW o FTP.

Los registros de recurso de servicio (SRV) permiten especificar de forma genérica la ubicación de los servidores para un servicio, protocolo y dominio DNS determinados. El formato de un registro SRV es el siguiente:

servicio.protocolo.nombre TTL clase SRV prioridad peso puerto destino

Donde:

- ✚ El campo **servicio**: especifica el nombre de servicio: http, telnet, etc.
- ✚ El campo **protocolo**: especifica el protocolo utilizado: TCP o UDP.
- ✚ **Nombre**: define el nombre de dominio al que hace referencia el registro de recurso SRV.
- ✚ Los campos **TTL y clase**: han sido definidos anteriormente.
- ✚ **Prioridad**: especifica el orden en que los clientes se pondrán en contacto con los servidores. Los clientes intentarán ponerse en contacto primero con el host que tenga el valor de prioridad más bajo, luego con el siguiente y así sucesivamente.
- ✚ **Peso**: es un mecanismo de equilibrio de carga.
- ✚ **Puerto**: muestra el puerto del servicio en el host.
- ✚ **Destino**: muestra el nombre de dominio completo para la máquina compatible con ese servicio.

Un ejemplo de registros SRV para los servidores Web del dominio admon.co., sería:

```
http.tcp.admon.com. IN SRV 0 0 80 www1.admon.com.  
http.tcp.admon.com. IN SRV 10 0 80 www2.admon.com.
```

11 Tipos de servidores de DNS.

Los servidores DNS pueden ser del tipo:

Primario o master: tiene autoridad sobre una zona primaria. Por tanto, en ese servidor, se pueden editar la zona o las zonas en las que actúa como servidor

Secundario o slave: tiene autoridad sobre una zona secundaria. Una zona secundaria de un servidor DNS obtiene la información de la zona primaria de otro servidor primario. A pesar de no poderse editar una zona secundaria, los clientes DNS pueden usar estos servidores DNS sin ningún problema.

Caché: no tiene autoridad sobre ninguna zona. Almacena temporalmente en caché las últimas consultas realizadas por el servidor. El tiempo de almacenamiento depende de la configuración que tenga establecida cada uno de los servidores DNS para los nombres que enviaron las resoluciones de las consultas.

Reenviador o forwarder: se considera así a un servidor DNS que ha sido designado por otro u otros servidores DNS para que se encargue de resolver nombres fuera del dominio en el que se encuentran.

Nota: Un servidor de nombres puede ser a la vez primario para algunas zonas, secundario para otras y caché. También puede ser adicionalmente reenviador para otro u otros servidores.

12 Instalación y configuración del servidor DNS en Linux.

El servidor más utilizado en sistemas operativos Linux, como ya sabes se llama **BIND**. Para instalar el servidor DNS en Ubuntu, puedes usar el gestor de paquetes "Synaptic" o instalar el paquete directamente mediante:

```
# apt-get install bind9
```

Los ficheros de configuración de bind en Ubuntu están en el directorio **/etc/bind**. Estos ficheros aparecen nada más instalar el servidor con una configuración básica e incluso con determinadas zonas ya configuradas. Los ficheros más importantes son los siguientes:

- ✚ **Ficheros db:** contienen tres zonas inversas: 0.in-addr.arpa, 127.in-addr.arpa y 255.in-addr.arpa. Estas tres zonas están siempre configuradas y hacen referencia al modo local: localhost. Contiene una zona vacía db.empty que se usa para crear nuestras zonas, la zona directa local (localhost) y, finalmente, el fichero db.root que contiene las direcciones IP de los trece servidores raíz.
- ✚ **Ficheros named.*:** hacen referencia a los ficheros de configuración del servidor.

12.1 Crear una zona primaria

Si configuras una zona Primaria para el dominio ejemplo.com, tienes que agregar una zona DNS a BIND9, el primer paso es editar **/etc/bind/named.conf.local** (recomendable hacer una copia de seguridad):

```
zone "ejemplo.com" {
    type master;
    file "/etc/bind/db.ejemplo.com";
};
```

Ahora utiliza un archivo de zona existente como una plantilla para crear el archivo /etc/bind/db.ejemplo.com:

```
sudo cp /etc/bind/db.local /etc/bind/db.ejemplo.com
```

Edita el nuevo archivo de zona /etc/bind/db.ejemplo.com y tienes que:

- ✚ Sustituir localhost. por el nombre (FQDN) de su servidor, dejando el "." final.
- ✚ Sustituye 127.0.0.1 por la dirección IP del servidor de nombres.
- ✚ Sustituye root.localhost por una dirección de correo electrónico válida, pero con un "." en lugar del símbolo usual "@", dejando de nuevo el "." al final.

También, crea un registro A para ns.ejemplo.com. El servidor de nombres en este ejemplo:

```
;
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA ns.ejemplo.com. root.ejemplo.com. (
    2019102800 ; Serial
    604800    ; Refresh
    86400     ; Retry
    2419200   ; Expire
    604800    ) ; TTL
;
@ IN NS ns.ejemplo.com.
@ IN A 127.0.0.1
ns IN A 192.168.1.10
```

Tienes que incrementar el número de serie cada vez que hagas cambios en el archivo de zona. Si haces múltiples cambios antes de reiniciar BIND9. Simplemente incrementa la serie una vez.

Ahora, puedes agregar registros DNS al final del archivo de zona.

Nota: A muchos administradores les gusta utilizar la última fecha de edición como la serie de una zona, así como 2018050100 que es yyyymmddss (donde ss es el Número de Serie).

Una vez que hayas hecho un cambio en el archivo de zonas, tendrás que reiniciar BIND9 para que los cambios tengan efecto:

```
# sudo service bind9 restart
```

12.2 Crear una zona inversa

Ahora que has configurado la zona y se resuelven nombres a direcciones IP, también se necesita una zona inversa. Esta zona permite al DNS resolver una dirección a un nombre.

Ahora vas a ver como se crea esa zona inversa modificando los ficheros de configuración de BIND, para ello:

Edita /etc/bind/named.conf.local y añade lo siguiente:

```
zone "1.168.192.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.192";
};
```

Reemplaza 1.168.192 con los tres primeros octetos de cualquiera que sea la red que estas utilizando. También, nombra el archivo de zona /etc/bind/db.192 apropiadamente. Debe de coincidir el primer octeto de tu red.

Ahora crea el archivo /etc/bind/db.192:

```
sudo cp /etc/bind/db.127 /etc/bind/db.192
```

Luego edita /etc/bind/db.192 cambiando básicamente las mismas opciones que en /etc/bind/db.ejemplo.com:

```
;
; BIND reverse data file for local loopback interface
;
$TTL 604800
@ IN SOA ns.ejemplpo.com. root.ejemplo.com. (
    2      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
    604800 ) ; TTL
;
@ IN NS ns.ejemplpo.com.
10 IN PTR ns.ejemplo.com.
```

El número de serie en la zona inversa debe incrementarse con cada cambio. Para cada entrada A que configuras en /etc/bind/db.ejemplo.com debes crear una entrada PTR en /etc/bind/db.192.

Después de haber creado el archivo de zona inverso reinicia BIND9:

```
# sudo service bind9 restart
```

12.3 Configurar un DNS secundario.

Una vez que has creado una zona primaria, es necesaria una zona secundaria para poder mantener la disponibilidad del dominio si el primario se vuelve no disponible. Recuerda que una zona secundaria contiene una copia de una zona primaria. Un servidor secundario actualiza la zona cada cierto tiempo por transferencia de zona desde el servidor primario de la zona.

Primeramente, en el servidor maestro primario, se necesita permitir la transferencia de la zona. Añade la opción allow-transfer a las definiciones de ejemplo de zona directa (Forward) e inversa (Reverse) en /etc/bind/named.conf.local:

```
zone "example.com" {
    type master;
```

```
file "/etc/bind/db.example.com";
allow-transfer { 192.168.1.11; };
};

zone "1.168.192.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.192";
    allow-transfer { 192.168.1.11; };
};
```

Nota: Reemplaza 192.168.1.11 con la dirección IP de tu servidor de nombres secundario.

A continuación, en el DNS secundario, instala el paquete bind9 de la misma forma que para el primario. Luego, edita /etc/bind/named.conf.local y añade las siguientes declaraciones para la zona directa (Forward) e inversa (Reverse):

```
zone "ejemplo.com" {
    type slave;
    file "/var/cache/bind/db.ejemplo.com";
    masters { 192.168.1.10; };
};

zone "1.168.192.in-addr.arpa" {
    type slave;
    file "/var/cache/bind/db.192";
    masters { 192.168.1.10; };
};
```

Nota: Reemplaza 192.168.1.10 con la dirección IP de tu servidor de nombres primario.

Reinicia BIND9 en el DNS secundario:

```
# sudo service bind9 restart
```

En /var/log/syslog deberías ver algo similar a:

```
slave zone "ejemplo.com" (IN) loaded (serial 6)
slave zone "100.18.172.in-addr.arpa" (IN) loaded (serial 3)
```

Un servidor secundario es aquél que no tiene los datos originales de la zona, tienen una copia. Estos servidores contactan con el primario, y así obtienen una copia.

12.4 Utilización de reenviadores externos

Un reenviador es un servidor DNS que se encarga de resolver consultas externas para otros servidores DNS. Esto básicamente quiere decir que si usamos un reenviador, este se dedica a resolver todos los nombres externos a nuestro dominio y, por tanto, resolver cualquier nombre de Internet.

Si queremos que nuestro servidor DNS utilice reenviadores externos como los que tienen las direcciones IP 195.235.113.3 y 195.235.96.9 simplemente tenemos que editar el archivo de configuración

/etc/bind/named.conf.options y escribir en la directiva forwarders que hay dentro de options {}, las direcciones de los reenviadores.

```
forwarders { 195.235.113.3; 195.235.96.9; };
```

13 Comandos relativos a la resolución de nombres

Cuando estés realizando la instalación y configuración de servidores DNS, puedes hacer uso de varios comandos relacionados con el servicio DNS que te permiten, entre otras cosas, comprobar el funcionamiento del servicio.

Los comandos más destacados relativos a DNS en sistemas Windows son:

```
nping
ipconfig
nslookup
```





Los comandos más destacados relativos a DNS en sistemas Linux son:

ping
hosts
nslookup
dig

13.1 nslookup

Nslookup es una herramienta de línea de comandos para solicitar consultas DNS a un servidor. Está disponible tanto en sistemas Windows como en sistemas Linux.

Hay tres formas de ejecutar nslookup:

-  **nslookup nombre.equipoDNS:** solicita la resolución DNS del nombre de equipo al primer servidor DNS.
-  **nslookup nombre.equipoDNS nombre.servidorDNS:** solicita la resolución DNS del nombre de equipo al servidor DNS indicado.
-  **nslookup:** muestra los nombres y las direcciones IP de los servidores DNS que hay configurados en la configuración de red del equipo y accede al modo interactivo.

Al entrar al modo interactivo cambia el prompt del sistema y se muestra como nuevo prompt el carácter ">" que nos indica que podemos ejecutar comandos de nslookup.

En el modo interactivo, para indicar que se obtengan los registros de un determinado tipo para consultas posteriores, se usa el comando de nslookup:

```
set q=tipoRegistro
```

Donde **tipoRegistro** es un tipo de registro de recursos como A o NS.

En el modo interactivo podemos consultar un nombre DNS (de equipo o de dominio) simplemente escribiendo el nombre. El resultado que se obtiene es una lista de todos los registros del tipo establecido que tienen como propietario el nombre introducido.

13.2 Dig

El comando dig es un comando incluido en sistemas Linux. Sirve para hacer consultas a servidores DNS y comprobar el resultado. Por ello, es adecuado para comprobar si un servidor DNS que estemos instalando y configurando funciona correctamente.

Para que dig nos devuelva los registros de un determinado tipo correspondientes a un nombre DNS, hay que ejecutar el comando con la sintaxis:

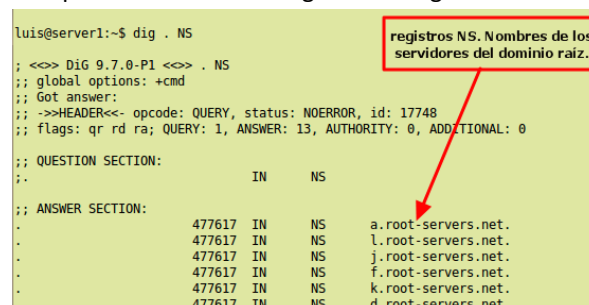
```
dig nombreDNS tipoRegistro
```

Ejemplos:

Para obtener una lista de los servidores DNS raíz:

```
#dig . NS
```

Y en Ubuntu se obtiene el resultado que se muestra en la siguiente imagen:



```
luis@server1:~$ dig . NS
; <<> DiG 9.7.0-P1 <<> . NS
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 17748
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 0



;; QUESTION SECTION:
; .                IN      NS

;; ANSWER SECTION:
.                477617 IN      NS      a.root-servers.net.
.                477617 IN      NS      l.root-servers.net.
.                477617 IN      NS      j.root-servers.net.
.                477617 IN      NS      f.root-servers.net.
.                477617 IN      NS      k.root-servers.net.
.                477617 IN      NS      d.root-servers.net.
```

13.3 Otros comandos.

ipconfig:

En relación con el servicio DNS, este comando permite:

-  Mostrar el contenido de la caché DNS del equipo ejecutando `ipconfig /displaydns`
-  Vaciar la caché DNS del equipo ejecutando `ipconfig /flushdns`

ping:

Al ejecutar este comando para enviar un "ping" hacia un ordenador para el que se especifique su nombre, el comando nos devuelve primero el resultado de la resolución DNS, es decir, nos devuelve la dirección IP correspondiente al nombre.

host:

Es un comando para sistemas Linux que, entre otras cosas, sirve para enviar consultas DNS al primer servidor DNS que tengamos establecido en la configuración de red de un equipo. Así, puede servir para probar los servidores DNS que hayamos instalado y configurado. La sintaxis de este comando para enviar consultas al primer servidor NS es

```
host -t tipoRegistro nombreDNS
```