

TEMA 3: Servicio de nombres de dominio

Módulo

Despliegue de aplicaciones web

para los ciclos

Desarrollo de aplicaciones web



Despliegue FP-GS; Tema3:ServicioDeNombresDeDominio

© Gerardo Martín Esquivel, Octubre de 2020

Algunos derechos reservados.

Este trabajo se distribuye bajo la Licencia "Reconocimiento-No comercial-Compartir igual 3.0 Unported" de Creative Commons disponible en <http://creativecommons.org/licenses/by-nc-sa/3.0/>

6.1 Necesidad.....	3
6.2 Espacio de nombres de dominio.....	3
6.3 Clasificación de dominios.....	5
6.3.1 Dominios de primer nivel.....	5
6.3.2 Dominios de segundo nivel.....	5
6.4 Gestión administrativa.....	6
6.4.1 Whois.....	6
6.4.2 Delegación DNS.....	6
6.4.3 Zonas DNS y dominios DNS.....	7
6.5 Funcionamiento DNS.....	7
6.5.1 El comando dig.....	8
El camino de la consulta.....	8
Consulta a un servidor concreto.....	9
Consulta inversa.....	10
6.5.2 nslookup.....	10
6.5.3 Consultas iterativas y recursivas.....	11
6.5.4 Resolución inversa.....	12
6.5.5 Almacenamiento de nombres en caché.....	12
6.6 Servidores DNS.....	13
6.7 Registros del DNS.....	13
6.8 DNS dinámico.....	16
6.8.1 DNS dinámico para administradores de zona.....	16
6.8.2 DNS dinámico para usuarios.....	16
6.9 Clientes DNS.....	17
6.9.1 En Windows.....	17
6.9.2 En Linux.....	18
6.9.3 El fichero hosts.....	19
6.10 Instalación y configuración del servidor DNS en Windows.....	19
6.10.1 Establecer IP, nombre y dominio.....	19
6.10.2 Instalación del servicio.....	20
6.10.3 Configuración del servicio.....	21
Ajustes en la configuración.....	22
6.10.4 Comprobando el servidor DNS.....	23
6.11 Instalación y configuración del servidor DNS en Linux.....	23
6.11.1 Configurar una zona primaria.....	23
6.11.2 Crear una zona secundaria.....	25
6.11.3 Crear una zona inversa.....	25
6.11.4 Opciones adicionales.....	26
6.11.5 Iniciar y detener el servicio.....	26

6.1 Necesidad

Como ya sabemos, cada máquina conectada a Internet se identifica con un número único que está compuesto de 32 dígitos binarios, conocido como **dirección IP**. Para facilitar el trabajo con direcciones **IP** las convertimos en grupos de 4 números decimales, pero aún así sigue siendo muy complicado recordar estas direcciones. Por este motivo se hace necesario poder acceder a los equipos utilizando nombres más fáciles de recordar, los dominios. Y lo único que necesitamos es que alguien **traduzca** esos nombres en las direcciones **IP** correspondientes.

Ese alguien que traduce los nombres en direcciones es el servicio **DNS** (Domain Name Server, Servidor de nombres de dominio). A través del servicio **DNS** podremos preguntar por un nombre de dominio y se nos devolverá la dirección **IP** correspondiente. También se puede hacer la tarea contraria: preguntar por una dirección **IP** y obtener el dominio correspondiente.

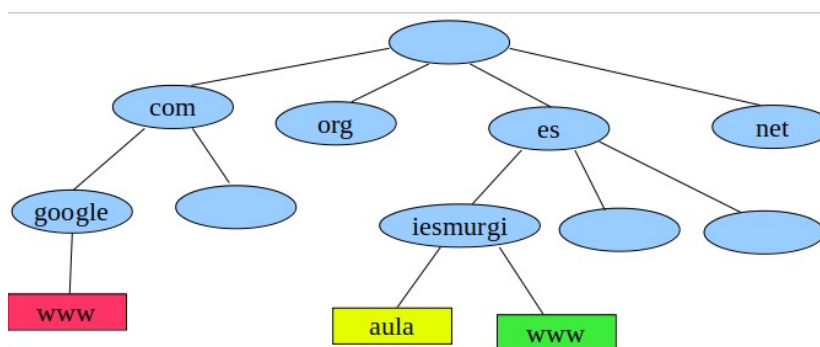
Nota: Aunque no lo sepas, tú utilizas el servicio **DNS** constantemente. Cada vez que escribes una URL en la barra de direcciones de un navegador, lo primero que se hace es una consulta al **DNS** para obtener la dirección **IP** y, una vez conocida, se le hace una petición al puerto **80** de la dirección obtenida para descargar las páginas web que ese equipo sirve.

6.2 Espacio de nombres de dominio

Un **sistema de nombres jerárquico** organiza los nombres en forma de árbol invertido (raíz arriba y abre las ramas hacia abajo) y permite controlar que no se repiten nunca dos nombres iguales. Si imaginamos los millones de máquinas conectadas a Internet y dejásemos que cada responsable de esas máquinas eligiese un nombre libremente, vemos como sería imposible que no se repitiesen los nombres. Al utilizar un sistema jerárquico, cada cual sólo se tiene que ocupar de que no se repitan nombres entre las máquinas que gestiona, pero no necesita saber los nombres de otras máquinas.

Un ejemplo de nombre jerárquico es el de las direcciones postales. Una entidad superior se preocupará de que en España no haya dos provincias con el mismo nombre. Más abajo habrá que cuidar que en la misma provincia no haya dos pueblos que se llamen igual. Los ayuntamientos ponen nombres a las calles y vigilarán que no haya dos calles con igual nombre, pero no debe importarles repetir el nombre de una calle con otro pueblo, porque la dirección postal completa siempre será distinta.

El **espacio de nombres de dominio** es el sistema de nombres jerárquico que utilizamos en Internet. Está organizado como una raíz sin etiqueta, seguida de los dominios genéricos de primer nivel y de los nombres que eligen las organizaciones conectadas a la red. Cada dominio puede subdividirse en otros dominios, hasta llegar al final de la estructura que son los nombres de los ordenadores.



Nota: Este árbol puede alcanzar un máximo de 127 niveles, aunque lo normal es usar 3 o unos pocos más. Cada una de las etiquetas puede tener un máximo de 63 caracteres.

El nombre completo de un equipo se construye comenzando en la raíz y escribiendo (hacia la izquierda y separando con puntos) los nombres de los dominios por los que pasamos hasta llegar al nombre del equipo. Observa que resulta imposible que el nombre completo de dos equipos sea igual a no ser que pongamos el mismo nombre a dos hermanos.

Ejemplo:

El nombre completo del equipo señalado en rojo sería:

www.google.com.

El nombre completo del equipo señalado en amarillo sería:

aula.iesmurgi.es.

El nombre completo del equipo señalado en verde sería:

www.iesmurgi.es.

Nota 1: Observa que los nombres anteriores *terminan en un punto*, porque después de ese punto está la etiqueta del nodo raíz, que es una etiqueta vacía. Es habitual que los nombres completos de dominio se escriban sin ese punto final. Los dominios que terminan en punto son dominios absolutos y los que no terminan en punto son dominios relativos. Se trata del mismo concepto de rutas absolutas (que empiezan con una barra -/-) y relativas (que no empiezan con barra) en los nombres completos de ficheros.

Nota 2: El punto final no es necesario para tareas habituales de usuario como buscar el dominio en la web. **Pero a la hora de configurar el servidor DNS habrá que recordar ese punto.**

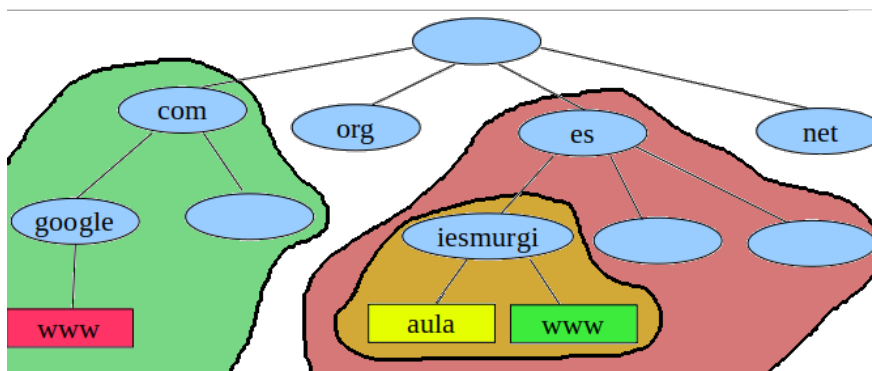
Es importante tener en cuenta que un dominio es todo un subárbol, es decir, el nodo que lleva ese nombre y todos los nodos que cuelgan de él.

El dominio de primer nivel **com** es todo el espacio señalado con fondo verde.

El dominio de segundo nivel **iesmurgi.es** es el espacio señalado con fondo marrón.

El dominio de primer nivel **es** ocupa el espacio señalado con fondo morado.

Fíjate que el dominio **es** incluye al dominio **iesmurgi.es**.



6.3 Clasificación de dominios

El control de las direcciones y dominios de Internet lo gestiona la **ICANN** (Internet Corporation for Assigned Names and Numbers, Corporación de Internet para la Asignación de Nombres y Números).

6.3.1 Dominios de primer nivel

Los dominios de primer nivel son los que cuelgan directamente de la raíz y los clasificamos según su uso en:

- Los **genéricos** fueron los primeros en existir (**com** para empresas, **org** para organizaciones sin ánimo de lucro, **int** para organizaciones internacionales, **edu** para organizaciones educativas, **gov** para el gobierno de EE.UU., **mil** para el ejército de EE.UU.). Actualmente los dominios de primer nivel genéricos se agrupan en dominios patrocinados y no patrocinados
 - ◆ Los **no patrocinados** (**com**, **org**, **net**,...) funcionan según las reglas globales de la **ICANN**.
 - ◆ Los dominios **patrocinados** (**gov**, **edu**, **mil**,...) que son gestionados por entidades concretas, aunque son creados previamente por la **ICANN**.
- Los **geográficos** son dominios de dos letras que hacen referencia generalmente a países (**es**, **it**, **de**, **us**, etc.) aunque no siempre (**eu**, Unión Europea). Son creados por el **ICANN** y las tareas de gestión de estos dominios están delegadas en los correspondientes gobiernos que generalmente cuentan con entidades que se encargan de ello. En España (dominio **es**) lo hace **Red.ES**.
- El dominio **arpa** es un dominio de primer nivel que se usa exclusivamente para la tarea de los **DNS** de traducir direcciones **IP** con dominios.
- Los **dominios reservados** son dominios de primer nivel que **no existen** y queda asegurado que **nunca existirán**. Son **test**, **example** e **invalid** que se utilizan para hacer pruebas y prácticas de manera que no interfieran en el servicio **DNS** real. Y el otro dominio reservado es **localhost** que apunta a la dirección **127.0.0.1** y por tanto hace referencia a la máquina desde la que se usa.

6.3.2 Dominios de segundo nivel

Los dominios de **segundo nivel** no patrocinados están disponibles para cualquier persona o entidad que los quiera comprar, siempre que no estén ya ocupados. De hecho ni siquiera es necesario pertenecer a los grupos que lo definen, así una empresa puede comprar un dominio **org** y un organización sin ánimo de lucro puede comprar un **com**. El registro de estos dominios sólo lo puede hacer la empresa que gestione el dominio de primer nivel donde colgará, pero la compra se puede hacer a través de muchos intermediarios que pueden tener precios muy distintos.

6.4 Gestión administrativa

La **ICANN** decide los dominios de primer nivel que van a existir, pero no los gestiona directamente, sino que delega esa gestión en una entidad por un periodo de tiempo que, después será renovado o asignado a otra entidad. Esa otra entidad puede delegar la gestión de los subdominios, de modo que la entidad que gestiona el dominio "**es**" puede crear tantos hijos como desee y la empresa que compra el dominio **miempresa.es** pasa a gestionar ese dominio y puede crear tantos hijos como desee. En el caso de los dominios geográficos la gestión se delega indefinidamente al gobierno correspondiente.

En el proceso de registro de un dominio intervienen tres términos que son tan parecidos que pueden crear confusión: **registro**, **registrador** y **registrante**.

- El **registro** es la entidad que ha recibido la cesión para gestionar un dominio de primer nivel. Por ejemplo, el dominio **.es** lo gestiona **Red.es**.
- El **registrador** es un intermediario que se dedica a vender los dominios. Puede haber varios registradores en cadena, mayoristas y minoristas, como en la venta de cualquier otro producto.
- El **registrante** es la persona o entidad que compra el dominio y será el propietario del dominio.

6.4.1 Whois

Es una base de datos distribuida que contiene los datos de los dominios. Se puede consultar desde algunas web. En **Linux** también desde la consola con:

```
whois dominio
```

Se pueden hacer consultas de dominios de cualquier nivel (**es**, **cat**, ...) y aparecerán los datos del registrante y del registrador.

Nota: Cuando compras un dominio puedes evitar que todos tus datos sean públicos utilizando un servicio de privacidad del **registrante** que ofrece sus datos en **whois**, aunque el dominio sigue siendo de tu propiedad.

6.4.2 Delegación DNS

Una de las tareas que tiene que realizar una entidad cuando se encarga de un dominio es mantener servidores **DNS** que permitan localizar a todos los equipos de su subárbol. Una vez delegado un subdominio, el gestor del dominio padre se puede olvidar de todo, sólo necesita conocer la lista de servidores **DNS** del subdominio para poder resolver los nombres de ese dominio.

La delegación de funciones se repite conforme bajamos el árbol:

- La **ICANN** delega la gestión del dominio **es** en la entidad **Red.ES**, que se encargará de mantener servidores **DNS** para localizar todas las máquinas de ese dominio e informará a **ICANN** de cuales son sus servidores **DNS**. Cuando los servidores de la **ICANN** reciban una pregunta sobre un equipo del dominio **es**, trasladarán la pregunta a los servidores de **Red.ES**.

- **Red.ES** delega la gestión del dominio **miempresa.es** a esa empresa, que se encargará de mantener servidores **DNS** para localizar todas las máquinas de ese dominio e informará a **Red.ES** de cuales son sus servidores **DNS**. Cuando los servidores de la **Red.ES** reciban una pregunta sobre un equipo del dominio **miempresa.es**, trasladarán la pregunta a los servidores de la empresa.

Esta tarea repetitiva se reproduce nivel a nivel, de modo que la única información que suministran los servidores **DNS** de primer nivel es cual es el servidor **DNS** del dominio que buscamos.

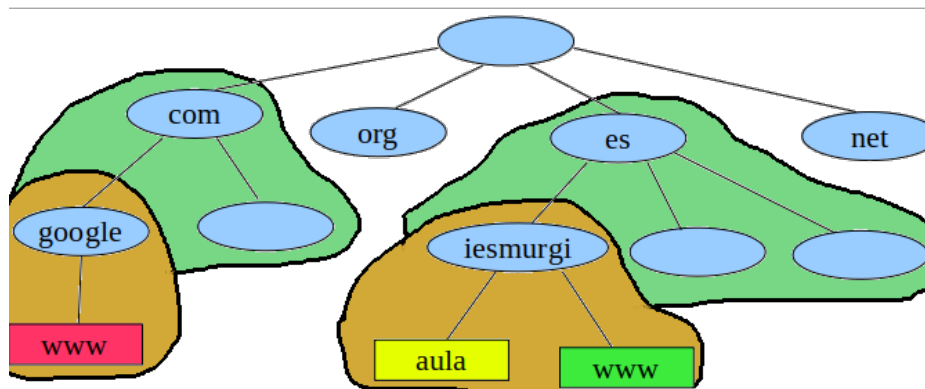
6.4.3 Zonas DNS y dominios DNS

Hemos visto que un dominio es el conjunto de todos los nodos que se agrupan bajo él, el subárbol completo.

La zona es solamente el conjunto de nodos cuyas direcciones se gestionan directamente, es decir, no están delegadas:

- La **zona raíz**: Sólo conoce las direcciones de los servidores **DNS** de los dominios de primer nivel.
- La **zona .es**: Sólo conoce las direcciones de los servidores **DNS** de sus dominios de segundo nivel, como **empresa.es**.
- La **zona .empresa.es**: Sólo conoce las direcciones **IP** de los equipos de la empresa.

En la imagen vemos la delimitación de las zonas. Si te fijas en la imagen anterior (la de los dominios) vemos como en **google.com** y en **iesabdera.es** la zona y el dominio son iguales, pero la **zona com** es más pequeña que el **dominio com** y la **zona es** es más pequeña que el **dominio es**.



Cuando una entidad se hace responsable de un dominio gestiona todo el dominio (a veces delegándolo) pero sólo mantiene las direcciones de su zona.

6.5 Funcionamiento DNS

Hay dos tipos de ordenadores en el sistema **DNS**: los clientes sólo hacen consultas y los servidores responden a las consultas y buscan direcciones **IP** de servidores desconocidos. Cuando se hace una consulta tiene lugar el siguiente proceso:

1. El cliente **DNS** hace una consulta (por ejemplo, **www.santana.org**) al servidor que tenga configurado como servidor **DNS** (normalmente será el servidor **DNS** de su **ISP**). Si este servidor conoce el dato lo suministra.
2. Si nuestro servidor **DNS** no conoce el dato le hará la pregunta al servidor **DNS** de la zona raíz que lo único que puede suministrar es la dirección del servidor de primer

nivel que conoce el dato (en nuestro ejemplo será la dirección del servidor **DNS** de la zona **org**).

3. Nuestro servidor **DNS** consulta al servidor de primer nivel indicado, que devolverá la dirección del servidor de segundo nivel que conoce el dato (en nuestro ejemplo será la dirección del servidor **DNS** la zona **santana.org**).
4. Nuestro servidor **DNS** consulta al servidor de segundo nivel indicado, que proporcionará la dirección que buscamos.

Dependiendo del número de niveles que tenga el dominio que buscamos, la operatoria puede continuar.

6.5.1 El comando **dig**

Nota: Las imágenes muestran la salida del comando **dig** en su **versión 9.9**. Se ha podido comprobar que las respuestas a los mismos comandos desde la **versión 9.11 (Ubuntu 18.04)** son más escuetas.

El comando **dig**, que puedes usar desde un terminal **Linux**, permite hacer consultas **DNS**:

dig dominio

Ejemplo 1:

En la imagen podemos ver que la respuesta al preguntar la dirección del dominio **www.google.com** es **216.58.211.68** y que nos aporta otros datos como los servidores de zona autoritativos y sus respectivas direcciones **IP**.

Esto significa que la zona que corresponde al dominio **google.com** está mantenida mediante 4 servidores de dominio que son los que se indican en la respuesta. Esos 4 servidores son **autoritativos** (quizá sería mejor traducción **autorizados**) en esa zona, es decir, sus respuestas sobre los dominios de esa zona son buenas, porque "saben de lo que hablan".

```
administrador@Laptop:~$ dig www.google.com

;<<>> DiG 9.9.5-9ubuntu0.3-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 46051
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                138     IN      A      216.58.211.68

;; AUTHORITY SECTION:
google.com.                    109078  IN      NS      ns1.google.com.
google.com.                    109078  IN      NS      ns2.google.com.
google.com.                    109078  IN      NS      ns4.google.com.
google.com.                    109078  IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
ns3.google.com.                132080  IN      A      216.239.36.10
ns2.google.com.                146657  IN      A      216.239.34.10
ns1.google.com.                139344  IN      A      216.239.32.10
ns4.google.com.                132080  IN      A      216.239.38.10

;; Query time: 179 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Tue Oct 20 19:30:12 CEST 2015
;; MSG SIZE rcvd: 195

administrador@Laptop:~$
```

EL CAMINO DE LA CONSULTA

Ejemplo 2: Si en el comando **dig** incluimos una petición detallada del camino seguido (**trace**):

dig www.google.com +trace

Tendremos una respuesta más larga:

Nota: En este ejemplo hemos eliminado gran parte de las líneas que obtenemos como salida del comando para hacer más claro el ejemplo.

Nota: Quizá sea necesario dirigir la petición al servidor DNS **8.8.8.8**: **dig @8.8.8.8 dominio +trace**

```
; <<>> DiG 9.9.5-9ubuntu0.3-Ubuntu <<>> www.google.com +trace
;; global options: +cmd
.                32983  IN      NS      c.root-servers.net.
.                32983  IN      NS      m.root-servers.net.
.                32983  IN      NS      e.root-servers.net.
;; Received 461 bytes from 127.0.1.1#53(127.0.1.1) in 7033 ms

com.             172800  IN      NS      l.gtld-servers.net.
com.             172800  IN      NS      g.gtld-servers.net.
;; Received 738 bytes from 202.12.27.33#53(m.root-servers.net) in 3390 ms

google.com.      172800  IN      NS      ns3.google.com.
google.com.      172800  IN      NS      ns4.google.com.
;; Received 664 bytes from 192.41.162.30#53(l.gtld-servers.net) in 800 ms

www.google.com.  300      IN      A       216.58.210.164
;; Received 48 bytes from 216.239.38.10#53(ns4.google.com) in 111 ms
```

En la respuesta hay cuatro partes correspondientes a las cuatro consultas realizadas:

- En la primera consulta se obtienen los servidores raíz, respuesta proporcionada por el propio equipo, o sea, **127.0.0.1**, porque los servidores raíz son conocidos por todos.

Nota: Los servidores raíz son 13 y están estratégicamente distribuidos por todo el mundo. El conjunto de esas 13 direcciones **IP** están en un fichero que se incluye en la instalación de todos los servidores **DNS**, de forma que todos las conocen. Cabe la posibilidad de que esas direcciones cambien en algún momento, así que sería buena idea actualizar la lista en nuestro servidor cada cierto tiempo. Puedes encontrar el fichero actualizado en <http://www.internic.net/zones/named.root>.

- En la segunda consulta se obtienen los servidores de la zona **com** y esta respuesta ha sido proporcionada por **m.root-servers.net**, que es uno de los servidores que hemos conocido gracias a la consulta anterior.
- En la tercera consulta se obtienen los servidores de la zona **google.com** y esta respuesta ha sido proporcionada por **l.gtld-servers.net**, que es uno de los servidores que hemos conocido gracias a la consulta anterior.
- En la cuarta consulta se obtienen la dirección de **www.google.com** que es lo que buscábamos y esta respuesta ha sido proporcionada por **ns4.google.com**, que es uno de los servidores que hemos conocido gracias a la consulta anterior.

Nota: Fíjate que en cada consulta se proporcionan varios servidores, pero sólo tenemos que preguntar a uno de ellos. Una forma habitual de elegirlos es según cuanto tardaron en responder la última vez.

CONSULTA A UN SERVIDOR CONCRETO

Podemos dirigir nuestra consulta a un servidor concreto usando el carácter arroba (@):

```
dig @servidorDNS dominio
```

Por ejemplo:

```
dig @ns1.google.com google.com
```

Está preguntando por el dominio **google.com** a su propio servidor de zona (**ns1.google.com**). Probablemente se obtenga una respuesta satisfactoria.

Sin embargo:

```
dig @ns1.google.com hotmail.com
```

Está preguntando por el dominio **hotmail.com** a un servidor de una zona distinta, concretamente de la zona **google.com**. Lo más probable es que nos informe de que no se acepta una consulta recursiva (el servidor de **google** no hará trabajo extra para **hotmail**).

CONSULTA INVERSA

El comando **dig** también permite hacer una consulta inversa, es decir, preguntar por el dominio que corresponde a una determinada dirección **IP**

```
dig -x direccionIP
```

Por ejemplo:

```
dig -x 202.12.27.33
```

devuelve la siguiente respuesta:

Si te fijas en la sección **ANSWER SECTION** verás que la dirección **IP** que hemos usado en la consulta corresponde al dominio **m.root-servers.net**.

Nota: **m.root-servers.net** es uno de los 13 servidores raíz que están a la cabeza del sistema del servicio de nombres de dominio.

```
administrador@administrador-TECRA-R950:~$ dig -x 202.12.27.33
; <<>> DiG 9.11.3-1ubuntu1.2-Ubuntu <<>> -x 202.12.27.33
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10471
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;; 33.27.12.202.in-addr.arpa.      IN      PTR
;; ANSWER SECTION:
33.27.12.202.in-addr.arpa. 86400 IN      PTR      m.root-servers.net.
;; Query time: 1055 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Tue Oct 16 16:17:28 CEST 2018
;; MSG SIZE rcvd: 86
```

6.5.2 nslookup

Este comando actúa como cliente **DNS** permitiendo hacer consultas sobre dominios o sobre direcciones **IP**.

Sintaxis:

```
nslookup dominio [servidorDNS]
nslookup direccionIP [servidorDNS]
```

Devuelve la dirección **IP** (o el dominio, respectivamente) que corresponde al dominio (o dirección **IP**) indicado según la respuesta que ofrece el servidor indicado. En el caso que omitamos el servidor **DNS** (parámetro opcional) hace la consulta al servidor por defecto (que será el que tenemos configurado en los parámetros **TCP/IP** de nuestro equipo).

Ejemplo: Consulta directa sobre un dominio

```
nslookup www.bbva.es
```

Devuelve la dirección **IP** asociada a ese dominio: **89.107.176.83**

Ejemplo: Consulta inversa (sobre una dirección IP)

```
nslookup 89.107.176.83
```

Observamos que devuelve varios dominios (recuerda que un equipo puede tener varios alias).

En las respuestas del comando **nslookup** se indica si la respuesta es autoritativa o no autoritativa. Una respuesta no autoritativa significa que servidor que ha contestado la consulta no es el encargado de resolver en ese dominio (no es el servidor de zona en ese dominio).

Las respuestas anteriores sobre **bbva.es** son **no autoritativas** porque el servidor consultado (el servidor por defecto) no es el encargado de resolver nombres en el dominio **bbva.es**

Ejemplos: Consulta de un dominio a su propio servidor de zona

```
nslookup www.bbva.es dnsbbva3.bbvamovil.com
```

En este caso volvemos a preguntar sobre el dominio **bbva.es**, pero dirigimos la pregunta a un servidor DNS muy concreto: su servidor de zona. La respuesta de ahora si es autoritativa.

Nota: Podemos averiguar el servidor encargado de resolver cada dominio con el comando

Nota: Cuando la respuesta no es autoritativa vemos el mensaje "**non-authoritative answer**". Cuando la respuesta es autoritativa, simplemente, no dice nada de eso.

Para entrar en el **modo interactivo de consultas al servidor por defecto**:

```
nslookup
```

Cuando lo usamos sin parámetros entra en modo interactivo permitiendo varias consultas sobre el servidor **DNS** por defecto. Para salir del modo interactivo debemos usar el comando **exit**.

Para entrar en el **modo interactivo de consultas a otro servidor**:

```
nslookup - servidorDNS
```

Entra en modo interactivo permitiendo varias consultas al servidor indicado. Para salir del modo interactivo debemos usar el comando **exit**.

Nota: Podemos averiguar el servidor encargado de resolver cada dominio con el comando

```
whois youtube.com
```

que está preinstalado en **Linux**. También podemos encontrar el servicio **whois** en la web, por ejemplo, haciendo una búsqueda por el término "**whois on-line**".

6.5.3 Consultas iterativas y recursivas

La **consulta iterativa** dará resultado inmediato sólo si el servidor consultado tiene la respuesta. Si no es así, sólo informará del siguiente servidor a quien preguntar. Este es el tipo de consulta que hace el servidor **DNS** de nuestro **ISP** en el ejemplo del apartado anterior. En una consulta iterativa el que pregunta está trabajando hasta dar con el resultado.

La **consulta recursiva** hace trabajar al servidor hasta dar con la respuesta o concluir que hay un error. Este es el tipo de consulta que hace el cliente **DNS** en el ejemplo del apartado anterior. En una consulta recursiva el que pregunta no trabaja, sólo espera el resultado.

No todos los servidores aceptan consultas recursivas, por ejemplo ninguno de la zona raíz la acepta. Es buena idea a la hora de configurar el servidor, no aceptar consultas recursivas de fuera de nuestra red local.

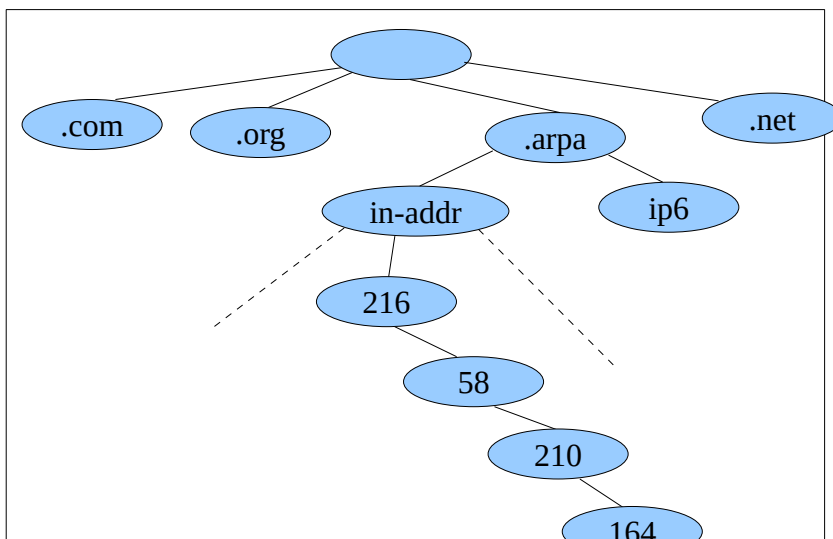
6.5.4 Resolución inversa

El sistema que hemos documentado permite obtener direcciones **IP** a partir de dominios. Cuando lo que queremos es obtener dominios a partir de direcciones **IP**, hablamos de **resolución inversa**.

Recordemos que el dominio de primer nivel **arpa** se usaba para funciones de configuración **DNS**. Este dominio tiene dos subdominios: **in-addr** y **ip6** que se usan para resolución inversa de los protocolos **IPv4** e **IPv6** respectivamente.

Para buscar el dominio de la dirección **IP 216.58.210.164** tendremos que preguntar por:

164.210.58.216.in-addr.arpa.



Observa que hay que escribir la dirección **IP** al revés y añadirle dos etiquetas resultando un dominio de 6 niveles.

Los dominios de la zona inversa también están delegados pero de forma distinta. Los rangos de direcciones **IP** son asignados a los **ISP** que se encargan de explotarlos y los asignan, según su criterio, de forma dinámica o fija. Cada **ISP** gestiona la zona **DNS** bajo el dominio **arpa**, cuyas direcciones le han sido asignadas y puede subdividirlas en zonas en función de sus intereses.

Hay que tener en cuenta que cuando compramos un dominio podemos gestionar nuestra zona por nosotros mismos, pero para que la zona inversa resuelva correctamente nuestra **IP** a nuestro dominio tendremos que ponernos en contacto con nuestro **ISP**, si es que ofrece ese servicio.

Puede ocurrir que haya discrepancias entre los resultados de la resolución directa y la inversa, porque son mecanismos independientes y los cambios realizados en uno no se reflejan en el otro.

6.5.5 Almacenamiento de nombres en caché

Cada vez que se realiza una consulta, la respuesta (del servidor autorizado) va acompañada del **TTL** (Time To Live, tiempo de vida) que es el tiempo durante el cual se considera válida esa respuesta (normalmente 1 o 2 días). El servidor **DNS** que ha hecho la consulta, guardará el resultado. Si volvemos a preguntar antes de que pase ese tiempo nos dará la misma respuesta sin necesidad de volver a hacer esa consulta. Si el tiempo ya ha expirado, nuestro **DNS** lanzará una nueva consulta.

Puesto que el servidor **DNS** de nuestro **ISP** atiende a muchos clientes, todas las consultas sobre el mismo dominio tienen la respuesta mientras dure el **TTL** sin necesidad de lanzar la consulta más allá, con lo que el tráfico se reduce considerablemente.

Además, el servidor **DNS** almacena todos los otros datos intermedios que obtiene para la consulta, todos esos que podemos ver cuando ejecutamos un **dig dominio +trace**. Por cierto, los números que aparecen junto a cada dato son los **TTL** en segundos para ese dato.

Los clientes **DNS** también pueden tener una caché con la misma finalidad.

6.6 Servidores DNS

Los servidores **DNS** son ordenadores que mantienen en ejecución un programa que permite aceptar consultas **DNS**. El puerto **53** es el que se usa habitualmente para este servicio. Los servidores **DNS** de zona deberían estar funcionando permanentemente.

Ya hemos visto que la traducción **DNS** de un dominio concreto la pueden resolver muchos servidores porque guardan en la caché datos de sus últimas consultas. Sin embargo, el único que podemos asegurar que tiene siempre la información correcta es el servidor de la zona a la que pertenece ese dominio. Se dice que la respuesta a una consulta que ha sido resuelta por el servidor de su dominio es una **respuesta autoritativa**.

Tipos de servidores **DNS**:

- **Servidores de zona:** Encargados de mantener las direcciones de su zona. Son los que dan respuestas autoritativas para todas las consultas sobre los equipos de su zona.
 - **Servidores de zona primarios:** (o **maestros**) En cada zona hay un único servidor primario. Guarda los datos originales y es sobre el que actúa directamente el administrador del servicio. Mantiene los datos aunque se reinicie.
 - **Servidores de zona secundarios:** (o **esclavos**) En cada zona hay uno o varios servidores secundarios. Guardan copia de los datos, obtenida periódicamente del servidor primario. Cuando se reinicia necesita obtener una copia de nuevo.

Nota: Los resultados de un servidor **secundario** son tan válidos (**autoritativos**) como los de un servidor **primario**, la única diferencia es que el administrador trabaja directamente sobre el primario. Siempre habrá un primario y uno o más secundarios, tantos más cuanto más grande sea la zona. Aconsejable que estén en redes distintas.

- **Reenviador:** (forwarders) Es un servidor **DNS** que recibe todas las peticiones de la red, con objeto de minimizar las consultas al exterior. Todos los demás servidores **DNS** de la red se configurarán para dirigir las consultas al reenviador.
- **Caché:** Sólo almacena en caché las consultas que hace con objeto de reducir consultas y no sobrecargar los servidores de zona. Por ejemplo, un **ISP** puede tener un servidor caché para no asignar a sus clientes directamente su servidor de zona.

6.7 Registros del DNS

La información que almacena la base de datos de un servidor **DNS** se estructura en registros. Cada registro contiene los siguientes datos:

- **Nombre de dominio:** que es la entrada a partir de la que se hace la búsqueda. Por ejemplo, **www.iesmurgi.org**. Debes recordar que es necesario terminar los dominios en punto (.) para que sean tratados como dominios absolutos.
- **Clase:** En nuestro caso siempre será **IN** (Internet).
- **Tipo:** El tipo de registro de que se trata, que puede ser **A**, **AAA**, **NS**, etc. Hay unos 30 tipos de registro.
- **TTL:** Tiempo de vida (en segundos) que se dará a la información de este registro. Si no aparece el **TTL** en el registro la información se servirá con el **TTL** por defecto.
- **Datos:** Es el dato de respuesta que el servidor devuelve.

Ejemplos de registros DNS:

pc01	IN	A		130.206.8.10
miempresa.es.	IN	MX	10	correo.miempresa.es.
miempresa.es.	IN	NS		dns1.miempresa.es.
miempresa.es.	IN	SOA		serv1.miempresa.es.
pepe.miempresa.es. (
2009082801				; Número de serie (serial)
86400				; Actualización (refresh)
7200				; Reintento (retry)
3600000				; Expiración (expire)
172800)				; TTL negativo

En este ejemplo aparecen 4 registros. Los tres primeros ocupan una única línea cada uno, el 4º registro tiene la parte de datos más extensa y se prolonga durante varias líneas más. Observa que algunos registros no tienen **TTL**, para esos registros tomará el valor **TTL** por defecto. Los comentarios van desde el símbolo punto y coma (;) hasta el final de línea. Si te fijas en la tercera columna verás que cada uno de los cuatro registros es de un tipo distinto (**A**, **MX**, **NS**, **SOA**). Existen muchos tipos de registro. Los más comunes son:

- **SOA:** (Start Of Authority) Este tipo de registro se usa para indicar el servidor principal de la zona. Contiene datos que permiten sincronizar el servidor primario con los secundarios. En la columna de datos de un registro **SOA** aparece el dominio del servidor principal de zona, el correo electrónico del responsable de zona y entre paréntesis 5 números, separados por punto y coma, que en este orden significan:
 - ◆ **Número de serie:** Lo usan los servidores secundarios para saber si la copia que tienen está actualizada o desfasada. Vale cualquier esquema de numeración que aumente con cada versión aunque se suele usar **YYYYMMDDnn**, donde **nn** es el número para las distintas actualizaciones de un mismo día.
 - ◆ **Actualización:** Es el tiempo en segundos que deben tardar los servidores secundarios en actualizar. Este número dependerá de la frecuencia de cambios, pero hay que elegirlo bien para evitar saturación y retraso en las actualizaciones.
 - ◆ **Reintento:** Es el tiempo en segundos que deben esperar los servidores secundarios para volver a intentar una actualización suponiendo que la última vez el acceso no fuese posible.
 - ◆ **Expiración:** Es el tiempo en segundos que un servidor secundario aguantará sin contacto con el servidor primario antes de autodestruirse. Es decir, borrará todos sus datos y dejará de considerarse autoritativo para la zona. Ya no será servidor de zona.
 - ◆ **TTL negativo:** Es el tiempo de vida en segundos de una respuesta negativa. Si una consulta **DNS** concluye que no existe un determinado dominio, se considera que esa respuesta es válida durante el tiempo establecido en **TTL negativo**.

Nota: Con el comando **dig** podemos consultar el registro **SOA** de un dominio con la sintaxis: **dig dominio SOA**

- **NS:** (Name Server) Estos registros indican los servidores de dominio de zona (primario y secundarios). Habrá tantos registros como servidores de nombres tengamos en la zona. Observa que los servidores de zona pueden estar situados dentro o fuera de la zona.

Nota: Cuando contamos que un dominio delegaba la gestión en sus subdominios decíamos aquello de "lo único que el dominio superior necesita saber son los servidores de nombres de la zona delegada". Pues bien eso significa que los registros **NS** de un dominio tienen que aparecer también en el dominio padre. Los registros **NS** de **miempresa.es** se repiten en los servidores del dominio **es**.

También será necesario propagar los registros **A** correspondientes a cada registro **NS** para que el dominio padre conozca, no sólo los nombres de los servidores de zona, sino también las direcciones **IP** correspondientes.

- **MX:** Servidor de correo de entrada para el dominio indicado. En este tipo de registros el dato de la cuarta columna no es el **TTL**, sino la **prioridad**. Si hay varios servidores de correo, la prioridad indica el orden en el que hay que contactar con ellos (número más bajo, mayor prioridad).

Si te fijas en ejemplo anterior, verás un registro **MX** que indica que si se recibe un correo electrónico a una dirección **persona@miempresa.es** debe llegar hasta el servidor **correo.miempresa.es** y puesto que la prioridad es 10 nos dice el orden en que tenemos que considerar este servidor supuesto que haya otros servidores de correo para el mismo dominio.

- **A:** Son los registros que usamos para enlazar un dominio con una dirección **IP**. En la primera columna aparecerá el nombre del equipo (podemos poner un nombre de equipo absoluto o uno relativo, en cuyo caso se entiende que es un equipo de nuestra zona). La dirección **IP** que aparece en el servidor será una **IP** pública naturalmente.

En el ejemplo anterior hay un registro de tipo **A** que asigna una **IP** al equipo **pc01**. Como es un nombre relativo (no acaba en punto) se sabe que se trata del equipo **pc01.miempresa.es**. y también se podría haber puesto este nombre absoluto.

- **AAAA:** Son los registros para enlazar dominios y direcciones **IPv6**.
- **CNAME:** Son registros para dar varios nombres a un mismo equipo (alias o apodo). En el ejemplo anterior definimos **pc01** mediante un registro **A**. Las siguientes líneas:

ftp	IN	CNAME	pc01
aula	IN	CNAME	pc01

añaden dos nombres más a ese ordenador, de modo que los nombres **pc01.miempresa.es**, **ftp.miempresa.es** y **aula.miempresa.es** apuntan al mismo equipo. Los alias no pueden usarse en la parte derecha de un registro, es decir, si queremos hacer referencia este equipo en un registro **NS** o **MX**, tendremos que usar **pc01**, pero no **ftp** ni **aula**.

- **PTR:** Los registros **PTR** se usan para crear la zona de búsqueda inversa, es decir, para poder obtener un dominio a partir de una dirección **IP**. El dato de la izquierda deberá ser el dominio **arpa** de la dirección **IP** (como **4.23.256.178.in-addr.arpa**. si fuese la dirección **178.256.23.4**) y el dato de la derecha será el dominio (absoluto) al que corresponde. Por ejemplo:

4.23.256.178.in-addr.arpa.	IN	PTR	ptkservices.net.
-----------------------------------	-----------	------------	-------------------------

6.8 DNS dinámico

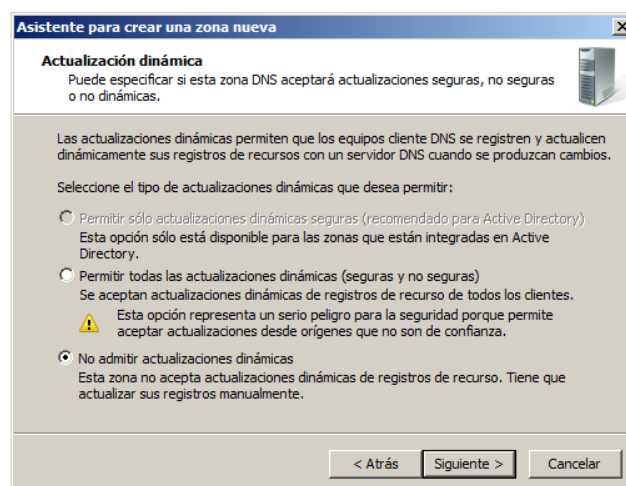
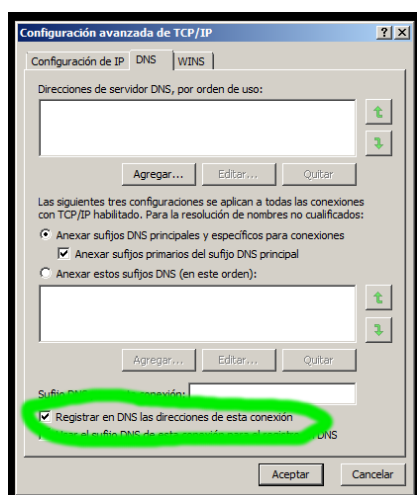
El sistema **DNS** está preparado para direcciones **IP** estáticas, pero la realidad es que con frecuencia las direcciones son dinámicas y esto puede generar algún problema hasta que se actualizan las bases de datos. Por ejemplo, si tenemos un servicio web en una dirección dinámica éste dejará de estar disponible a través del dominio (que es como lo usarán los usuarios) durante el tiempo **TTL** que se estableció la dirección como válida.

Hay dos formas de solucionar este problema: una adecuada para administradores de zona y otra adecuada para usuarios.

6.8.1 DNS dinámico para administradores de zona

Consiste en configurar el cliente **DNS** de los equipos cuyas **IP** pueden cambiar para que cuando ocurran esos cambios informen a su servidor de forma automática. Y también habrá que configurar el servidor **DNS** para que acepte ese mecanismo de actualización.

- **Configurar el cliente:** (Imagen de la izquierda) En **Windows**, los clientes están configurados por defecto para informar al servidor **DNS** de su nombre y dirección **IP**. Puedes modificarlo en **Propiedades TCP/IP / Configuración Avanzada**.



- Configurar el servidor:** (Imagen de la derecha) En **Windows**, al crear una nueva zona, se nos preguntará si admitimos este mecanismo de actualización. Recuerda que generalmente no es una buena idea.

6.8.2 DNS dinámico para usuarios

Existen empresas que ofrecen el servicio de **DNS** dinámico (**dyndns**, **no-ip**) y normalmente tiene un coste económico. En el equipo que cambiará de **IP** se instala un software que informa de esos cambios y el proveedor del servicio modificará los datos de zona. No forma parte del **DNS** estándar pero es muy útil para usuarios sin conocimientos.

A veces este servicio se ofrece mediante un dominio de tercer nivel, por ejemplo, **midominio.no-ip.org**, pero también se puede conseguir el **DNS** dinámico con nuestro propio dominio.

6.9 Clientes DNS

Cuando contratamos una conexión a Internet con un **ISP**, éste nos proporciona las direcciones **IP** de sus servidores **DNS**, normalmente dos, que debemos configurar como principal y alternativo.

Nota: Recuerda que es importante que uses los **DNS** que te proporciona tu **ISP** y, si cambias de proveedor, deberías cambiar a sus **DNS**. Probablemente sólo aceptará todas tus consultas si eres cliente.

Si en el contrato se incluye un router que gestiona el propio **ISP**, ese router ya vendrá configurado con los servidores **DNS** de tu proveedor, de modo que en el cliente de tu ordenador sólo tendrás que ajustarlo para que reciba esos datos de forma automática por **DHCP**.

Si ya dispones de un router propio serás tú el que tenga que anotar las **DNS**. Puedes hacerlo en el router para que las proporcione vía **DHCP** o puedes hacerlo en cada equipo de la red.

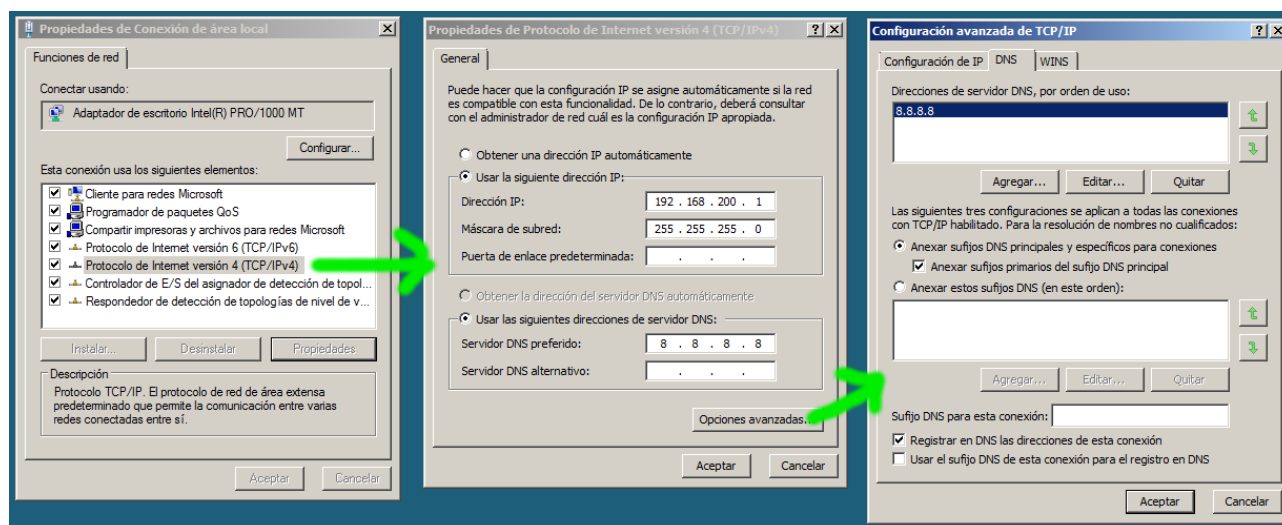
6.9.1 En Windows

INICIO/PANEL DE CONTROL/CENTRO DE REDES

selecciona la conexión deseada y, a continuación:

PROPIEDADES/PROTOCOLO DE INTERNET VERSIÓN 4 (TCP/IPv4)/PROPIEDADES

Ya te debe sonar esta ventana porque es la misma donde establecíamos la dirección **IP**, la máscara y la puerta de enlace. Fíjate que en la parte inferior también puedes indicar si las **DNS** se obtienen automáticamente o si, por el contrario, las quieres establecer manualmente.



Si pulsamos **Opciones avanzadas**, en la pestaña **DNS** tenemos:

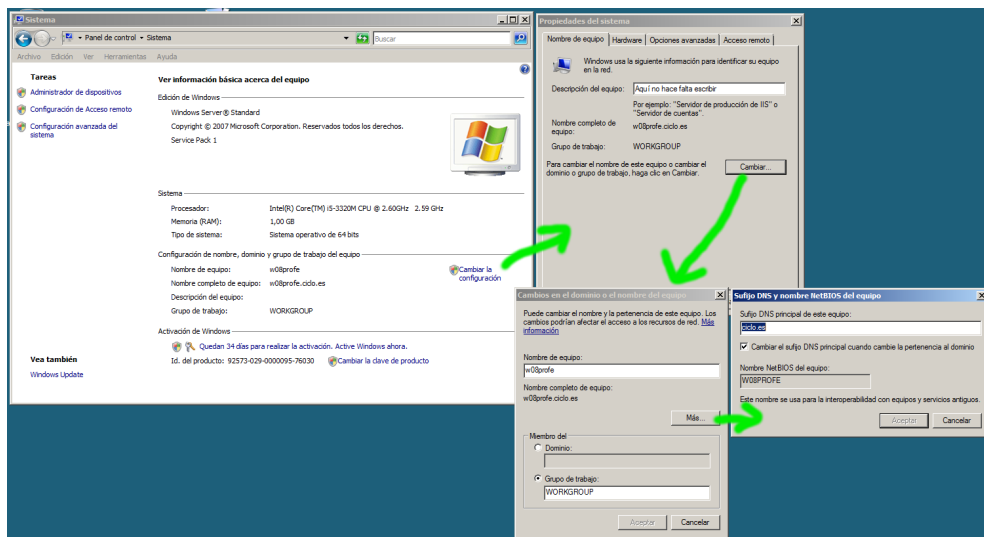
- La posibilidad de añadir más servidores (más de 2) en la parte superior de la pantalla.
- **Sufijos:** Los sufijos son dominios para que cuando el equipo no sea capaz de encontrar un equipo por su nombre, pruebe añadiéndole esos dominios de búsqueda como sufijos. Por ejemplo, si hemos añadido los sufijos **insti.es** y **ciclos.net**, cuando hagamos referencia a un equipo llamado **pc01** se probará con **pc01**, **pc01.insti.es** y **pc01.ciclos.net**.

- **"Registrar en DNS las direcciones"**: Este registro es el que mencionábamos en el apartado **"DNS dinámico para administradores de zona"** que había que hacer en el cliente **DNS**.

Para completar la configuración del cliente **DNS** será necesario establecer un nombre de equipo y el dominio al que pertenece. El nombre ya lo pusimos al instalar **Windows**, aunque ahora lo podemos cambiar.

INICIO/PANEL DE CONTROL/SISTEMA/CAMBIAR CONFIGURACIÓN

En la pestaña nombre de equipo aparece una descripción (ojo! eso no es el nombre). Para cambiar el nombre debemos pulsar **"Cambiar"** y escribirlo en **"Nombre de equipo"**. Si pulsamos en **"Más"** podemos también añadir un sufijo al nombre del equipo.




Nota: No es necesario configurar un sufijo para acceder a Internet, pero puede ser una buena idea si estamos ante ordenadores de una empresa bajo un mismo dominio **DNS**.

6.9.2 En Linux

Pulsa el icono de **"Configuración del Sistema"**



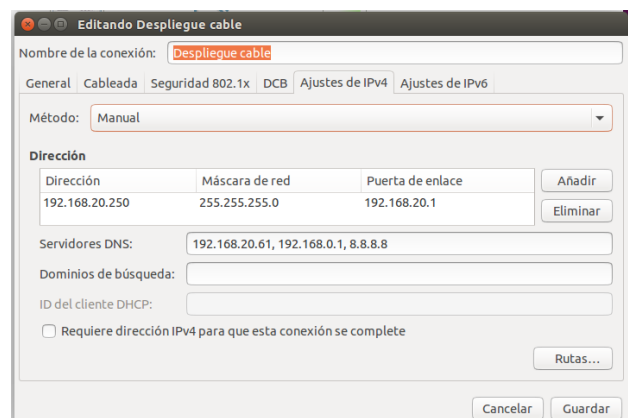
, el icono de **"Red"**, elige la

conexión adecuada y pulsa su flechita . Pulsa el botón **"Configuración"** y aparecerá la ventana de la imagen:

Si en **"Método"** eliges una de las opciones:

- **"Manual"**
- **"Sólo direcciones automáticas"**

podemos establecer las **DNS** separadas por comas y dominios de búsqueda separados por comas. Los dominios de búsqueda harán que cuando el equipo no sea capaz de encontrar un equipo por su nombre, pruebe añadiéndole esos dominios de búsqueda como sufijos.



También puedes establecer esos datos modificando manualmente el fichero `/etc/resolv.conf` con la siguiente sintaxis:

```
search dominio1 dominio2
nameserver ipDNS1
nameserver ipDNS2
```

Para indicar el dominio al que pertenece un servidor puedes escribir (en el mismo fichero):

```
domain dominio
```

Nota: No se recomienda trabajar directamente sobre este fichero porque determinadas aplicaciones como los clientes **DHCP** lo sobrescribirán cada cierto tiempo.

6.9.3 El fichero hosts

El fichero **hosts** es un fichero de texto que contiene una lista de direcciones **IP** con sus correspondientes nombres. Es algo así como un **DNS** rudimentario pero su importancia está en que tiene prioridad sobre las consultas **DNS**, es decir, que sólo se hará una consulta **DNS** si no hay una respuesta en este fichero.

Ejemplo de fichero hosts:

```
192.168.10.204 Laptop
127.0.1.1 Laptop
127.0.0.1 localhost
192.168.14.113 virtual-VirtualBox
```

En los sistemas **Linux**, este fichero se encuentra en:

```
/etc/hosts
```

En los sistemas **Windows**, se encuentra en:

```
...Windows\System32\drivers\etc\hosts
```

Nota: El fichero `/etc/hostname` guarda el nombre del ordenador.

6.10 Instalación y configuración del servidor DNS en Windows

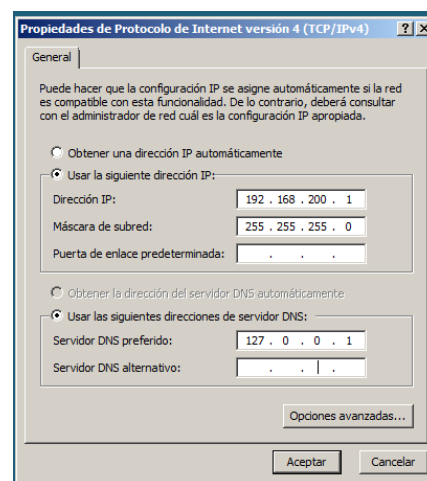
La instalación del servidor **DNS** se hará en tres pasos:

1. Establecer **IP**, nombre y dominio.
2. Instalación del servicio.
3. Configuración del servicio.

6.10.1 Establecer IP, nombre y dominio

Para establecer una dirección **IP** fija, tal y como hemos hecho otras veces, accedemos a la pantalla de la imagen.

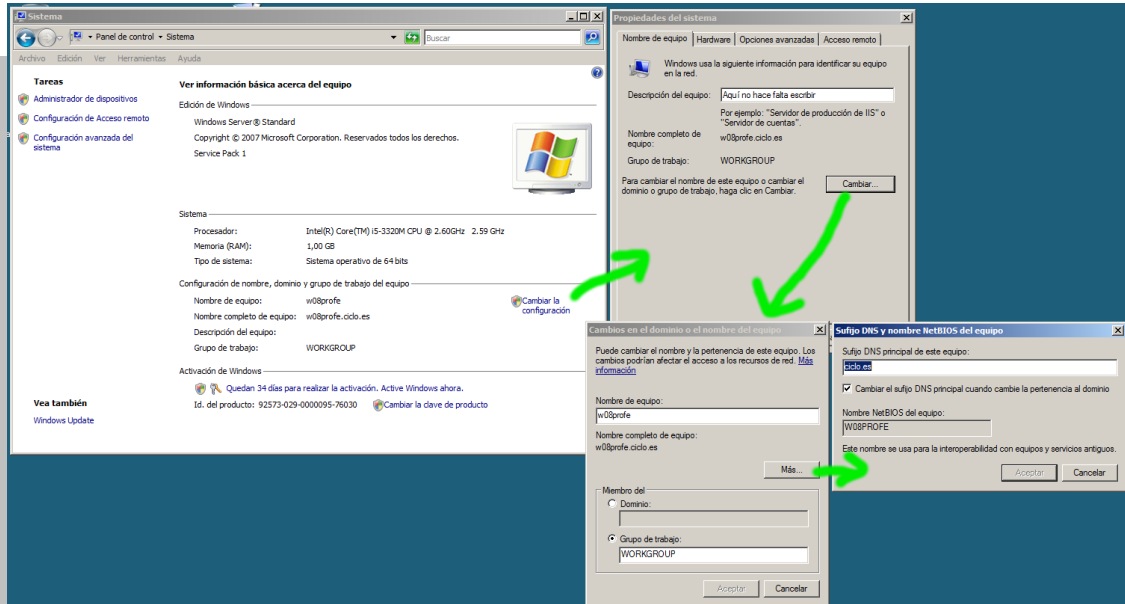
Observa que en esta ocasión decimos que el **DNS** preferido está en **127.0.0.1**, que hace referencia al propio equipo en el que estamos instalando un servidor **DNS**.



Para establecer un nombre de equipo y un dominio:

INICIO/PANEL DE CONTROL/SISTEMA/CAMBIAR CONFIGURACIÓN

Se abre la ventana **"Propiedades del sistema"**. En la pestaña **"nombre del equipo"** la descripción es opcional, pero pulsamos el botón **"cambiar"**, ponemos un **"nombre de equipo"**, pulsamos el botón **"más..."** y escribimos el sufijo (dominio) al que pertenece.

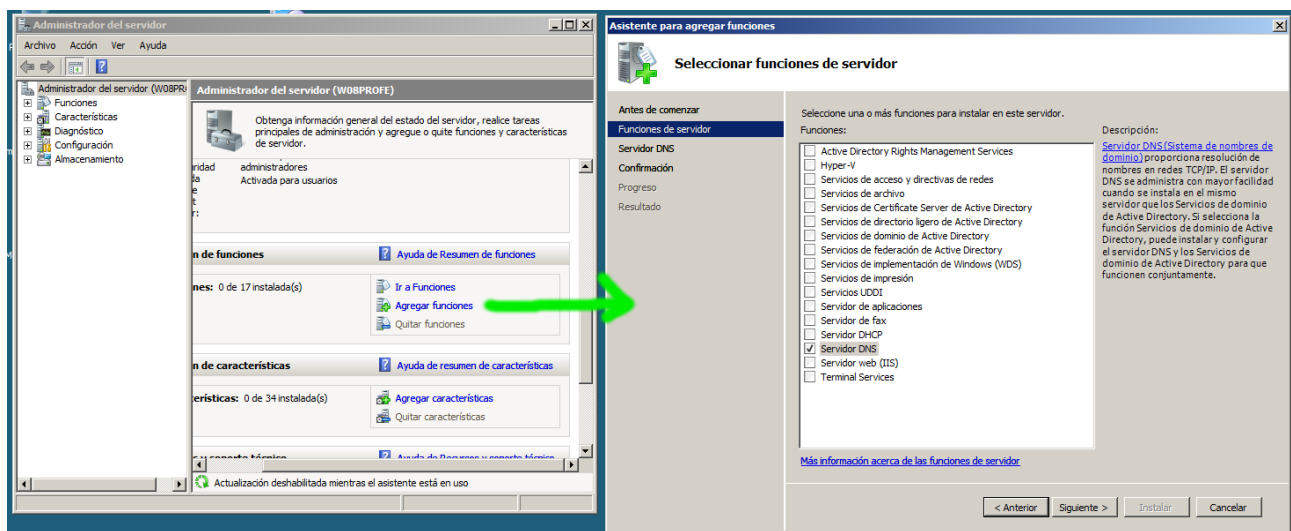


Nota: Al final pedirá reiniciar que es necesario.

6.10.2 Instalación del servicio

INICIO/HERRAMIENTAS ADMINISTRATIVAS/ADMINISTRACIÓN DEL SERVIDOR/AGREGAR FUNCIONES/SIGUIENTE

Seleccionamos **"servidor DNS"**, **Siguiente** e **Instalar**.

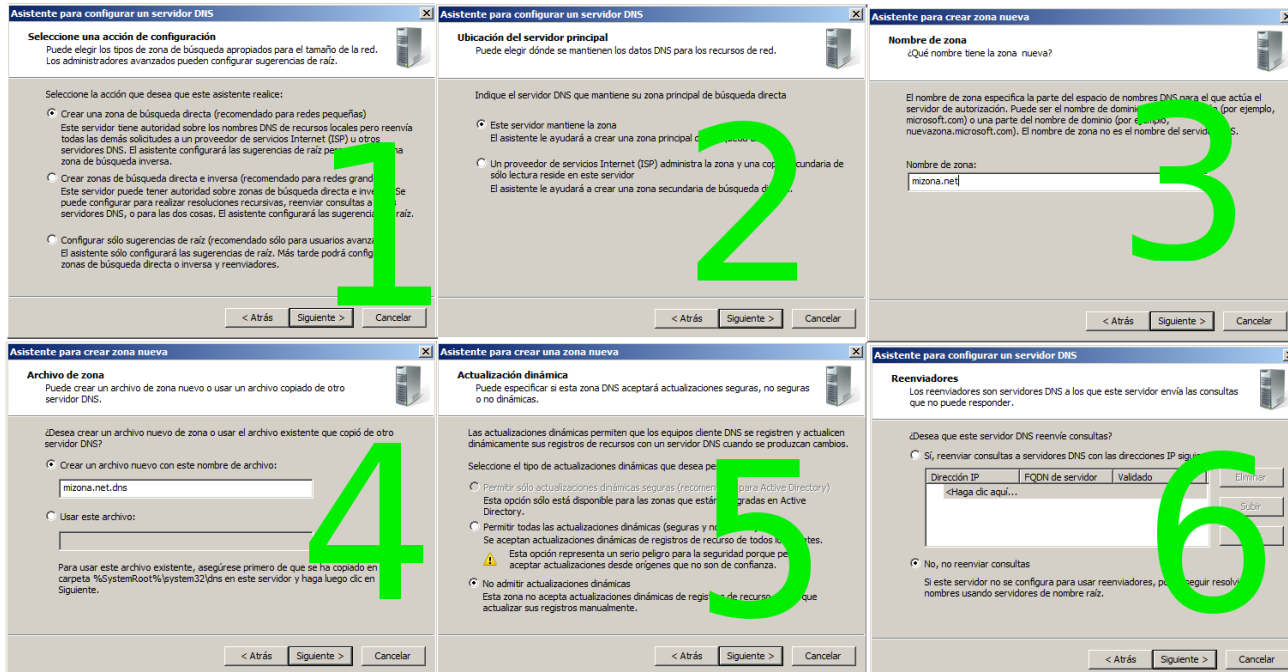


6.10.3 Configuración del servicio

INICIO/HERRAMIENTAS ADMINISTRATIVAS/DNS

Y a continuación:

ACCIÓN/CONFIGURAR SERVIDOR DNS



Al crear un **servidor DNS primario** debemos elegir (ver imagen 1) entre:

- zona de búsqueda directa, o
- zona de búsqueda directa e inversa

Nota: Esto último sólo si también tenemos delegada la gestión de la zona inversa. Debes recordar que administrar la zona significa administrar la zona directa. Para administrar la zona inversa es necesario que nos lo delegue nuestro **ISP** suponiendo que preste ese servicio.

En la siguiente pregunta (ver imagen 2) estamos eligiendo si el servidor que creamos es:

- primario ("**Este servidor mantiene la zona**"), o
- secundario ("**Un ISP administra y este es una copia secundaria de sólo lectura**")

El "**nombre de zona**" (ver imagen 3) es el dominio que vamos a administrar.

En la imagen 4 nos muestra el nombre del fichero que propone para guardar los datos. Lo aceptaremos a no ser que tengamos algún motivo para cambiarlo.

En la siguiente pantalla (ver imagen 5) marcamos "**no admitir actualizaciones dinámicas**" porque eso puede facilitar la tarea del administrador pero es un agujero de seguridad. Los clientes podrían actualizar los registros **A** suplantando a otros equipos.

Uso de **reenviadores** (ver imagen 6). Aquí colocamos la dirección del servidor en el que delegamos las consultas que no somos capaces de resolver (normalmente será el **DNS** proporcionado por el **ISP**). Si no ponemos reenviadores, nuestro servidor contactará directamente con los servidores raíz comenzando el proceso estudiado.

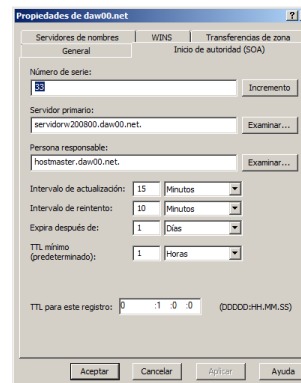
AJUSTES EN LA CONFIGURACIÓN

Si seleccionamos una zona y nos vamos a:

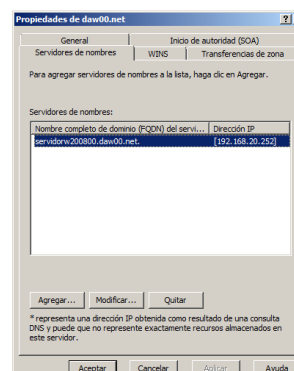
Acción/Propiedades

podremos ajustar otros parámetros de la zona. Tenemos una ventana con 5 pestañas:

- Pestaña **SOA**: Ya conocemos los datos que contiene el registro **SOA**. En esta ventana podemos modificarlos. El número de serie en los sistemas **Windows** comienza en 1 y se actualizará automáticamente con cada actualización.

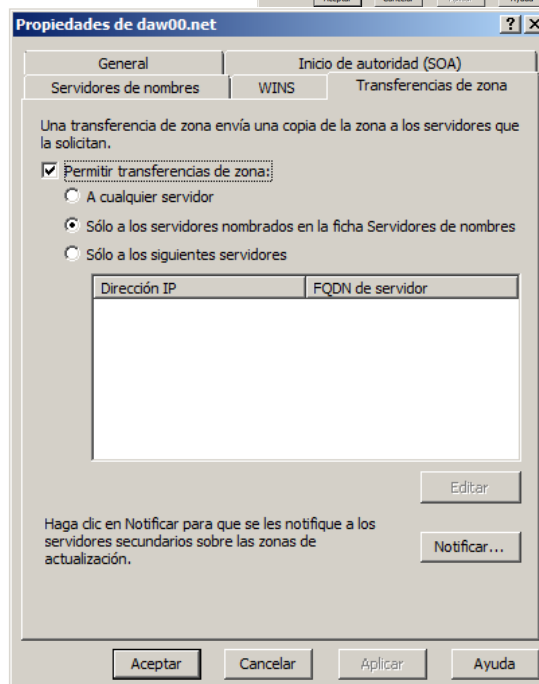


- En la pestaña "**Servidores de nombres**": Indicamos los servidores de zona (el principal, que es este y los secundarios, si los hubiera).



- En la pestaña "**Transferencia de zona**": se indican los equipos a los que se informa de todo el contenido. Normalmente sólo a los otros servidores de zona que tienen que hacer una copia completa. El resto de clientes y servidores **DNS** sólo preguntará por una entrada determinada. La opción "**a cualquier servidor**" sólo se debería usar en tareas de depuración de errores.

Al configurar un servidor secundario sólo hace falta indicar la **IP** de su servidor primario. Un servidor puede ser al mismo tiempo primario de una zona y secundario de otra, pero nunca primario y secundario de la misma.



6.10.4 Comprobando el servidor DNS

Podemos comprobar la zona directa y la inversa con **nslookup**. Se trata de un comando que está disponible tanto desde el terminal de **Windows** como del de **Linux**.

Deberíamos de hacer consultas

- desde el propio servidor
- desde la red local (cualquier equipo de la red distinto al servidor)
- desde fuera de la red, con las herramientas de página web que ofrecen:

➤ **dig** (<http://www.kloth.net/services/dig.php>)

➤ **nslookup** (<http://www.kloth.net/services/nslookup.php>)

6.11 Instalación y configuración del servidor DNS en Linux

La instalación se reduce a instalar el paquete **bind9** (como **root**):

```
apt-get install bind9
```

Una vez instalado, aparece la carpeta **/etc/bind/**, donde se encuentran los ficheros de configuración:

- **db.0**, **db.127** y **db.255** son zonas inversas preconfiguradas que hacen referencia a **localhost**.
- **db.empty** es una zona directa vacía que será el punto de partida de nuestras propias zonas.
- **db.local** es una zona directa preconfigurada para **localhost**.
- **db.root** contiene las direcciones **IP** de los 13 servidores raíz.
- **named.conf** no se debe tocar aunque bastaría con escribir en él toda la configuración.
- **named.conf.options** para las configuraciones globales.
- **named.conf.local** para crear nuevas zonas.

Nota: El proceso del servidor **DNS** se llama **named** y lo ejecuta el usuario **named**.

6.11.1 Configurar una zona primaria

En el fichero **/etc/bind/named.conf.local** con la siguiente sintaxis:

```
zone "midominio.dom" {  
    type master;  
    file "/etc/bind/primario/db.midominio.dom";  
    allow-transfer { none; };  
};
```

Nota: En estos ficheros se tomarán por comentarios todas las líneas que comienzan con doble barra (**//**).

- **type master** porque es una zona primaria.
- Con **file** indicamos el fichero que contiene los datos de la zona. Aún no existe, lo tendremos que crear.

- Tras **allow-transfer** indicamos las direcciones de los equipos a los que se les aceptará una solicitud de **transferencia de zona** completa (los secundarios). Es importante, porque si no se indica, se permite transferencia de zona a cualquier **IP**. Para indicar que no se permita a ninguna dirección IP escribimos **none**;

Tenemos que:

- crear la carpeta **/etc/bind/primario**

```
mkdir /etc/bind/primario
```

- copiar allí el fichero **db.empty**

```
cp /etc/bind/db.empty /etc/bind/primario/db.empty
```

- renombrarlo como **db.midominio.dom** (el directorio y el fichero tienen que tener permiso de lectura para el grupo y el grupo tiene que ser **bind**)

```
mv /etc/bind/primario/db.empty /etc/bind/primario/db.midominio.dom
```

Nota: Los dos últimos comandos se pueden resumir en uno:

```
cp /etc/bind/db.empty /etc/bind/primario/db.midominio.dom
```

El fichero debe tener un aspecto similar al siguiente:

```
; BIND reverse data file for empty rfc1918 zone
;
; DO NOT EDIT THIS FILE - it is used for multiple zones.
; Instead, copy it, edit named.conf, and use that copy.
;
$TTL      86400
@ IN SOA localhost. root.localhost. (
        1      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        86400 ) ; Negative Cache TTL
;
@ IN NS      localhost.
aaa IN A      192.168.200.245
fff IN CNAME  aaa.midominio.dom.
```

- Los comentarios en este fichero son las líneas que comienzan por punto y coma (;).
- El registro **SOA** (es obligatorio) ya venía en el fichero del que hemos partido.
- El siguiente registro es para aclarar quien es el servidor de zona del dominio. Fíjate que se trata de un registro **NS** y que la parte de la izquierda debería ser la zona en la que trabaja. El carácter arroba (@) se puede usar como sustituto de la zona (en nuestro caso **midominio.dom**).
- El tercer registro es un registro del tipo **A** que asigna el nombre **aaa** (recuerda que el nombre completo será **aaa.midominio.dom**.) a la dirección **192.168.200.245**.
- El cuarto registro es del tipo **CNAME** (alias) y asigna el nombre **fff** al mismo equipo.

Nota: con **\$TTL** se asigna el valor por defecto de **TTL** (en segundos).

6.11.2 Crear una zona secundaria

En el mismo **fichero named.conf.local** con la siguiente sintaxis:

```
zone "otrodominio.dom" {
    type slave;
    masters { direccionIP; };
    file "/etc/bind/secundario/db.otrodominio.dom";
    allow-transfer { none; };
};
```

- **type slave** porque es secundario.
- Usamos **masters** para indicar los servidores desde los que transferir la zona completa, normalmente sólo el primario.
- Tras **allow-transfer** indicamos **none** porque no vamos a permitir la transferencia de zona desde el secundario.

Tenemos que:

- Crear la carpeta **/etc/bind/secundario**

```
mkdir /etc/bind/secundario
```

- La carpeta debe tener permiso de escritura para el grupo para que pueda el servidor crear el fichero:

```
chmod g+w /etc/bind/secundario
```

6.11.3 Crear una zona inversa

En el fichero **/etc/bind/named.conf.local**:

```
zone "200.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/primario/200.168.192.in-addr.arpa";
    allow-update { none; };
    allow-transfer { none; };
};
```

Y en el fichero **/etc/bind/primario/200.168.192.in-addr.arpa**:

```
@ IN SOA dns.midominio.dom. root.localhost. (
    1      ; Serial
    604800 ; Refresh
    86400  ; Retry
    2419200 ; Expire
    86400 ) ; Negative Cache TTL
;
200.168.192.in-addr.arpa. IN NS localhost.

245 IN PTR aaa.midominio.dom.
200 IN PTR bbb.midominio.dom.
```

6.11.4 Opciones adicionales

Otras opciones se pueden especificar en el fichero `/etc/bind/named.conf.options` que tiene un aspecto como el siguiente:

```
options {  
    ...  
    allow-recursion { 192.168.200.0/24; 127.0.0.1; ::1; };  
    ...  
    forwarders { direccionIP; };  
    forward only;  
    ...  
}
```

- Con la directiva **allow-recursion** indicamos los equipos a los que se atenderá si hacen una consulta recursiva. Al resto de equipos sólo se les atenderá con consultas iterativas. Recuerda que esto es lo mejor para evitar que equipos ajenos a tu empresa sobrecarguen su servidor **DNS**. En este ejemplo hemos indicado que se aceptan consultas recursivas de todos los equipos de nuestra red (**192.168.200.0/24**) y de **localhost** (que se ha expresado tanto en la versión **IPv4** como en la **IPv6**).
- Con la directiva **forwarders** se está indicando que este servidor será un **reenviador**. Es decir, cuando no conozca la respuesta a una consulta, siempre la enviará a la dirección indicada y no hará otra tarea. Recuerda que si tenemos varios servidores **DNS** en nuestra empresa será útil que todos usen un mismo **reenviador** que será el único que realmente haga las consultas para evitar trabajo extra.

Nota: Los ficheros `/var/log/messages` y `/var/log/syslog` mantienen un registro del servidor que será útil para solucionar problemas.

6.11.5 Iniciar y detener el servicio

Para iniciar el servicio **DNS**:

```
service bind9 start
```

Para detenerlo:

```
service bind9 stop
```

Para reiniciarlo:

```
service bind9 restart
```