

## Tema 3. Servicio de Nombres de Dominio. DNS

1. Características. ....	3
2. Componentes.....	3
3. Espacio de nombres de dominio. ....	4
3.1. Nombres de dominio relativos y absolutos.....	4
3.2. Administración de nombres de dominio.....	4
3.3. Delegación de dominios.....	5
3.4. Registros de dominio. ....	5
4. Servidores de nombres. ....	6
4.1. Tipos de servidores de nombres.....	7
5. Clientes DNS (resolvedores).....	8
6. Mecanismo de resolución.....	9
6.1. Respuestas en caché.....	10
7. Correspondencias inversas.....	10
8. Registros de recursos.....	11
8.1. Tipos de registros.....	12
8.2. Registros pegamento (Glue Record).....	13
9. Transferencias de zona. ....	14
10. DNS dinámico (DDNS, Dynamic DNS) ....	14
11. Seguridad DNS.....	15
12. Servidores DNS en Linux. ....	15
13. Servidores DNS en Windows. ....	21
14. Configuración de clientes.....	22
14.1. Clientes Linux.....	22
14.2. Clientes Windows.....	23
15. Herramientas de consulta. ....	24
16. Anexo 1. El resolutor en Ubuntu 17 y posteriores. ....	25



## Tema 3. Servicio de nombres de dominio. DNS.

El servicio de resolución de nombres usado en las redes TCP/IP es el servicio DNS *Domain Name System* o Sistema de Nombres de Dominio. Este servicio permite identificar de una forma más sencilla a un equipo mediante un nombre, en lugar de usar la identificación numérica de la dirección IP.

### 1. Características.

En las redes TCP/IP, como es Internet, no es fácil recordar las direcciones IP de los equipos. Es mucho más cómodo usar y recordar nombres que secuencias de números.

Para facilitar el uso de los servicios, recursos y equipos de una red se creó un sistema de nombres que mediante un servicio de resolución de nombres permite asociar nombres con direcciones numéricas. De forma simplificada podemos decir que un servicio de nombres almacena direcciones y sus nombres correspondientes.

La operación que hay que realizar para conocer la dirección IP de un equipo a través de su nombre se denomina *resolución de nombre*.

Los sistemas de nombres se clasifican en planos y jerárquicos. En los sistemas de nombre planos, los nombres se usan sin ninguna estructura u organización que permita clasificarlos de alguna manera. Por ejemplo, los números de matrícula de los alumnos de un centro y su correspondencia con los nombres de dichos alumnos. (Los nombres NetBIOS que usa Windows para identificar equipos son un espacio de nombres plano). El mayor inconveniente es que los nombres deben ser únicos, sin repetición.

Los sistemas de nombres jerárquicos usan nombres que se agrupan o clasifican bajo algún criterio. Esta cualidad permite una gestión más simple, permitiendo que los nombres se puedan repetir y con la posibilidad de realizar una gestión de forma distribuida. Ejemplo de ello son los nombres en un sistema de archivos y también como veremos, el servicio de resolución de nombres DNS.

El servicio DNS es un servicio de registro y consulta de información, la cual se almacena en una base de datos distribuida en numerosos equipos. Estos equipos que almacenan una parte de dicha información de denominan *servidores de nombres*.

La información a la hora de distribuirla entre los diversos servidores de nombres se organiza mediante un esquema de nombres jerárquico. Este esquema jerárquico es lo que se denomina *espacio de nombres de dominio*. Así **cada servidor de nombre gestiona sólo una parte del espacio de nombres de dominio. Dicha parte se denomina dominio y es un subárbol del espacio de nombres de dominio.**

Los clientes DNS se dedican a preguntar a los servidores de nombres, los cuales responden usando para la comunicación entre ellos el protocolo DNS.

El servicio DNS no sólo permite asociar un nombre de dominio con una dirección IP, además permite almacenar otras informaciones, como por ejemplo qué equipos ofrecen un determinado servicio, cuál es el servidor de correo del dominio o qué equipos son fuentes de *malware* o de *spam*.

### 2. Componentes.

El servicio DNS se basa en el modelo cliente-servidor y está formado por los componentes siguientes:

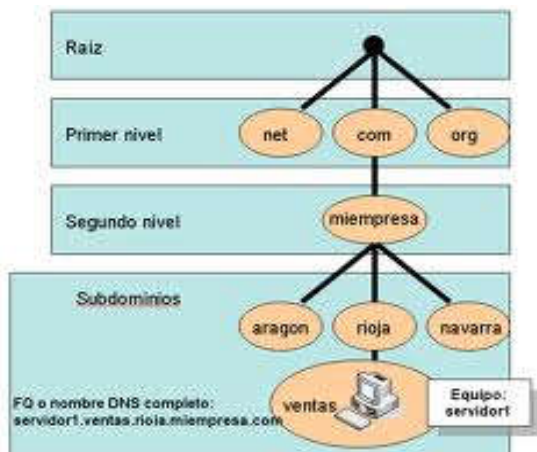
- Espacio de nombres de dominio. Lo forma la totalidad del conjunto de nombres, estructurados de forma jerárquica, que permite identificar equipos o servicios de red.
- Base de datos DNS. Es una base de datos distribuida que contiene la información del espacio de nombres del dominio. La base de datos DNS se organiza en *zonas*, en las cuales los datos se registran mediante los llamados *registros de recursos (RR)*.
- Servidores de nombres. Los servidores DNS guardan parte de la base de datos DNS en las llamadas *zonas* y son capaces de responder a las preguntas relativas a la información que almacena en sus zonas. Normalmente un servidor de nombres guarda información de una sola zona, es decir la información correspondiente a esa parte del espacio de nombres o dominio, pero también puede registrar la información de varios dominios, cada uno en su zonas correspondiente.
- Clientes DNS o resolutores. Son los encargados de realizar las preguntas a los servidores de nombres y ofrecer las respuestas a las aplicaciones que las solicitan.
- Protocolo DNS. Conjunto de reglas y normas que usan los servidores y clientes DNS para dialogar.

### 3. Espacio de nombres de dominio.

Como hemos comentado, el espacio de nombres de dominio lo forma el conjunto de nombres que permite identificar equipos o servicios de red, estructurados de forma jerárquica.

Estos nombres se denominan *nombres de dominio* y están formados por una serie de caracteres separados por puntos. Por ejemplo: *google.es.*, *info.*, *servidor.aula.izv.*

El espacio de nombres de dominio se puede representar mediante una estructura jerárquica arborescente, donde cada nodo del árbol se separa de otro mediante el carácter punto.



Los nombres de dominio no pueden superar los 255 caracteres distribuidos en como máximo 127 niveles, los cuales pueden contener como máximo 63 caracteres.

Los nombres de dominio siempre terminan con un punto, ya que el árbol de nombres de dominio empieza en el dominio “.” o dominio *raíz*. Los dominios que cuelgan del dominio raíz se denominan *dominios de primer nivel* o *dominios de nivel superior* (TLD, Top Level Domains). Los que cuelgan de los dominios de primer nivel se denominan *dominios de segundo nivel*. Normalmente a los dominios de tercer nivel e inferiores se les suelen denominar *subdominios*, aunque sólo es cuestión de nomenclatura.

#### 3.1. Nombres de dominio relativos y absolutos.

Cuando se hace referencia a un dominio, podemos usar su nombre relativo o absoluto.

Si usamos nombres *relativos* debemos conocer cuál es el dominio superior para saber a qué nombre nos estamos refiriendo exactamente. Si uso como nombre *puesto7* a no ser que el dominio de contexto sea por ejemplo *aula.izv.* no sabré exactamente a qué equipo estoy haciendo referencia.

Los nombres absolutos están formados por todos los nombres de nodos separados por puntos desde el nodo correspondiente hasta el nodo raíz “inclusive”. Por ejemplo: *puesto7.aula.izv.*

Los nombres absolutos se denominan *nombres de dominio perfectamente cualificados* (FQDN, Fully Qualified Domain Names).

Los nombres de dominio no sólo sirven para hacer referencia a un equipo en el espacio de nombres de dominio. También sirven para hacer referencia a un subárbol del espacio de nombres del dominio, es decir para hacer referencia a un nodo, y a todos los nodos descendientes que cuelgan de éste.

Por ejemplo, el dominio *es.* hace referencia a todos los nodos (subdominios y equipos) que cuelgan del dominio de primer nivel *es*. El dominio *aula.izv.* hace referencia a todos los nodos que cuelgan del dominio de segundo nivel *aula.izv.*

Por lo tanto un dominio permite hacer referencia a un conjunto de equipos y subdominios que se agrupan según un criterio. Uno de los criterios más usados, y que suele causar confusión, es el de los equipos que pertenecen a una misma red. Por ejemplo, en nuestro caso el dominio *aula.izv.* agrupa a todos los equipos del aula, los cuales están en la misma red (192.168.210.0/24), pero esto no quiere decir que todos los equipos que pertenezcan a una misma red deben pertenecer al mismo dominio. De hecho se pueden usar otros criterios de agrupación: equipos de una misma empresa (aunque los equipos estén en redes distintas...), equipos que prestan servicios, equipos que estén en una misma ubicación física, etc.

#### 3.2. Administración de nombres de dominio.

En Internet, la administración y la organización del espacio de nombres de dominio se realiza a través de diversas empresas y organizaciones coordinadas por la ICANN (*Internet Corporation for Assigned Names and Numbers* <http://www.icann.org/>).

El ICANN tiene la misión de que Internet sea funcional y entre otras cosas, de administrar el dominio raíz, asignar bloques de direcciones IP y mantener un registro de los dominios de nivel superior (TLD).

Antiguamente, InterNIC (<http://www.internic.net/>) asumía las responsabilidades que ahora gestiona el ICANN; pero actualmente como organización asociada al ICANN, es la encargada proporcionar al público información relativa a los servicios de registro de nombres de dominio de Internet.

Los dominios de primer nivel (TLD) se clasifican por parte de la ICANN en los siguientes tipos:

- Genéricos (gTLD). Usan un nombre relacionado con el propósito o el tipo de organización que lo va a utilizar. Éstos a su vez se clasifican en:
  - Patrocinados (sTLD). Existe una organización que lo patrocina. Ejemplos: "info", "asia", "edu", "gov", etc.
  - No patrocinados (uTLD). Operan con unas reglas comunes establecidas por el ICANN. Por ejemplo: "com", "net", "org", "mil", "gov", etc.
- Geográficos (ccTLDs). Usan dos letras en función del país. La gestión de estos dominios es delegada por el ICANN a organizaciones propias de cada uno de los países denominadas *operadoras de registro*. Ejemplo de dominios geográficos son: "es", "uk", "fr". En España, Red.es (<http://www.red.es>) es la organización delegada por el ICANN para la gestión del dominio "es.", y del registro de dominios de segundo nivel dentro de "es."
- arpa. El dominio "arpa" lo gestiona directamente la ICANN y sirve a través de los subdominios "in-addr.arpa" y "ip6.arpa" para poder efectuar la resolución inversa de direcciones IP.
- Reservados. Son dominios de primer nivel reservados sólo para pruebas y documentación. Por ejemplo: "test", "example" y "localhost".

### 3.3. Delegación de dominios.

Ya comentamos que el servicio DNS se basa en la administración distribuida de la información del espacio de nombres de dominio entre múltiples servidores de nombres. Esto se consigue mediante la delegación.

La delegación permite que una organización que administra un dominio, ceda la administración de sus subdominios (de uno, de varios, o de todos) a otras organizaciones. Por ejemplo, la ICANN, administradora del dominio raíz, delega en otras organizaciones los dominios de primer nivel (TLD) y se denominan operadoras de registro (*registry operators*). En España Red.es administra el dominio de primer nivel "es.", el cual delega a otras organizaciones los dominios de segundo nivel.

Por ejemplo, el dominio "ugr.es." es delegado por Red.es a la Universidad de Granada, la cual puede a su vez delegar o no sus subdominios a otras organizaciones. Así el subdominio "etsit.ugr.es." es delegado por la Universidad de Granada a la ETS de Informática y Telecomunicaciones.

El hecho de que un dominio se divida en subdominios no implica que tenga que ser delegado. Por ejemplo la UGR puede crear el subdominio "derecho.ugr.es." pero no delegar su administración.

### 3.4. Registros de dominio.

El registro de un dominio consiste en reservar un nombre durante un tiempo para poder crear subdominios, y asociar al dominio y a sus subdominios direcciones IP.

El registro de nombres de segundo nivel lo realizan empresas u organizaciones acreditadas denominados agentes registradores (*registrant*) los cuales asesoran a los clientes que quieren registrar un dominio, y tramitan las solicitudes operando como intermediarios entre los clientes y las operadoras de registro de primer nivel.

Así, en España, Red.es es la operadora de registro de primer nivel para el dominio "es.", y aunque puede registrar directamente dominios de segundo nivel a los clientes, éstos utilizan normalmente los servicios de los agentes registradores debido a que suelen ofrecer otros servicios complementarios (alojamiento web, servidores de correo, etc.) y a precios mucho más competitivos.

En los sitios web pertenecientes a ICANN e InterNIC podemos obtener la lista de empresas registradoras acreditadas que ofrecen servicios de registro de dominios genéricos; y en Red.es las empresas registradoras acreditadas para el dominio "es."

## 4. Servidores de nombres.

Los servidores de nombres o servidores DNS almacenan una parte de la base de datos del espacio de nombres guardando información sobre nombres de dominio y respondiendo a los clientes DNS u a otros servidores DNS. Los servidores escuchan por dos puertos: 53/TCP y 53/UDP.

La parte de información del espacio de nombres de dominio que mantienen los servidores de nombres se denomina zona. La información de una zona se almacena en archivos de texto o en bases de datos, dependiendo del tipo de servidor. Cuando un servidor de nombres contiene información de una zona se dice que es *autorizado (authoritative) para esa zona*.

Los ficheros de zona almacenan básicamente sus datos mediante registros de recursos. Dependiendo del tipo de información que se asocie con un nombre de dominio se utiliza un tipo de registro de recurso.

Por ejemplo, un servidor DNS para los equipos de un aula almacenaría la información de la zona, que podríamos denominarla “*aula.izv.*” y en dicha zona, mediante registros, se definirían los nombres que cuelgan de “*aula.izv.*” como por ejemplo “*puesto1.aula.izv.*”, “*puesto2.aula.izv.*”, etc.

Aunque posteriormente lo examinaremos con más detalle, el servidor **BIND** (*Berkeley Internet Name Domain*), el servidor DNS más comúnmente usado en Internet y un estándar de facto, registra los datos en formato texto. Los registros de recursos más usuales son:

- NS. Indica los servidores de nombres de dominio DNS que tienen autoridad para una zona.
- A. Asocia un nombre de dominio con su dirección IP.
- CNAME. Asigna otro nombre de dominio o alias a un nombre de dominio ya existente.

Ejemplo de un archivo de zona de resolución directa del dominio “*aula.izv.*”, que se almacena en un servidor DNS que está en el equipo con IP 192.168.210.1 cuyo nombre es *puesto1.aula.izv.*

```
...
aula.izv.          IN      NS      puesto1.aula.izv.
puesto1.aula.izv.  IN      A        192.168.210.1
puesto2.aula.izv.  IN      A        192.168.210.2
puesto3.aula.izv.  IN      A        192.168.210.3
www.aula.izv.      IN      CNAME    puesto2.aula.izv.

;subdominio sri.aula.izv. delegado
sri.aula.izv.      IN      NS      puesto10.sri.aula.izv.
puesto10.sri.aula.izv. IN    A        192.168.210.10

;subdominio bd.aula.izv. no delegado
puesto15.bd.aula.izv. IN    A        192.168.210.15
puesto16.bd.aula.izv. IN    A        192.168.210.16
puesto17.bd.aula.izv. IN    A        192.168.210.17
...
```

El archivo de zona del dominio *aula.izv.* indica mediante el registro "NS" que el equipo *puesto1.aula.izv.* es un servidor DNS autorizado para el dominio *aula.izv.* Con el primer registro de tipo "A" establece que dicho equipo, *puesto1.aula.izv.* usa la dirección IP 192.168.210.1. A continuación los otros registros tipo "A" asocian un nombre de dominio con una dirección IP, y con CNAME indicamos otro nombre de dominio (*www.aula.izv.*) para el equipo *puesto2.aula.izv.*

También podemos comprobar cómo se ha delegado el subdominio *sri.aula.izv.* en otro servidor DNS con dirección IP 192.168.210.10 y cuyo nombre es *puesto10.sri.aula.izv.* Este servidor será autorizado para el subdominio *sri.aula.izv.* y almacenará el archivo de zona de dicho subdominio.

El archivo de zona del subdominio delegado *sri.aula.izv.* podría ser:

```
...
sri.aula.izv.      IN      NS      puesto10.sri.aula.izv.
puesto10.sri.aula.izv. IN    A        192.168.210.10
puesto11.sri.aula.izv. IN    A        192.168.210.11
puesto12.sri.aula.izv. IN    A        192.168.210.12
...
```

Por otro lado, el subdominio *bd.aula.isv*. no se ha delegado ya que la zona también almacena los RR del subdominio. Por lo tanto el servidor DNS queda autorizado para los dominios *aula.isv* y *bd.aula.isv*.

Es común el error de considerar los términos zona y dominio como sinónimos. Un dominio es un subárbol del espacio de nombres del dominio. Los datos asociados a los nombres de un dominio pueden estar almacenados en una o varias zonas, las cuales pueden estar distribuidas en uno o varios servidores DNS.

En nuestro ejemplo anterior podemos comprobar cómo los nombres de dominio *aula.isv*. se distribuyen en dos archivos de zona, el fichero de la zona *aula.isv* y el fichero de la zona *sri.aula.isv*.

Además, un servidor de nombres puede tener autoridad sobre varias zonas. Por ejemplo, un mismo servidor puede ser autorizado para las zonas *aula.isv*. y *taller.isv*.

#### 4.1. Tipos de servidores de nombres.

Un servidor autorizado para una zona puede dejar de funcionar, o bien estar colapsado por la carga de trabajo. Para evitar estos problemas y mejorar el servicio de resolución de nombres, DNS permite almacenar una misma zona en varios servidores DNS. Podemos imaginarnos que el servicio de búsquedas de Google no iría tan rápido si hubiera un único servidor autorizado para el dominio *google.com*.

Veremos que estas situaciones se resuelven mediante *zonas maestras* y *zonas esclavas*.

Según la función que realizan los servidores de nombres se pueden clasificar en:

- Servidor **maestro** (*master*). Define una o varias zonas para las que es autorizado. El administrador es el responsable de los archivos de zona, añadiendo, modificando o borrando nombres de dominio.

Si un cliente DNS le pregunta por un nombre de dominio para el que está autorizado, consultará en los archivos de su zona y le responderá. Si no está autorizado, buscará la información en otros servidores DNS si así está configurado, o responderá que no sabe la respuesta.

- Servidor **esclavo** (*slave*). Define una o varias zonas para las que es autorizado, pero obtiene los archivos de zona de otro servidor autorizado para la zona (el maestro u otro esclavo), realizando lo que se denomina *transferencia de zona*. Los archivos de zona del servidor esclavo no se pueden editar, por lo que las modificaciones deben realizarse en el maestro que realiza la transferencia.

Un servidor DNS puede ser maestro para una o varias zonas y a la vez esclavo de otras. Además pueden existir varios servidores esclavos para una misma zona.

Con los servidores esclavos se puede repartir la carga de trabajo entre varios servidores y disminuir los fallos que se puedan producir en el servicio.

- Servidor **caché**. Si un servidor DNS recibe una pregunta sobre un nombre de dominio para la que no está autorizado, podrá preguntar, si está configurado así, a otros servidores DNS. Si además este servidor actúa como caché, las respuestas obtenidas de los otros servidores las podrá almacenar durante un tiempo (TTL, *Time To Live*). De esta forma, cuando un cliente u otro servidor DNS le formule una pregunta, consultará primero en su memoria caché, por si ya tuviera la respuesta.

Los servidores llamados *solo caché*, son los que no tienen autoridad sobre ningún dominio y siempre realizan las preguntas a otros servidores para resolver las peticiones, guardando las respuestas en su memoria caché. En redes extensas y con muchos equipos, es adecuado tener un servidor DNS que actúe de esta forma ya que se logra disminuir significativamente el tráfico en la red.

- Servidor **reenviador** (*forwarding*). Cuando un servidor DNS no tiene respuesta a una petición de resolución, puede si así está configurado, realizar la pregunta a otros servidores (en un orden que luego explicaremos), hasta obtener una respuesta; o bien puede trasladar directamente la pregunta a otros servidores DNS (*forwarders*), para que sean éstos los que se encarguen de resolverla.

Por lo tanto, un servidor configurado como reenviador traslada las preguntas que no puede resolver a otro servidor DNS al que se le traslada dicha consulta. Así se consigue disminuir el tráfico de peticiones DNS en Internet generado por los equipos de una red (se encargan de resolver los *forwarders*) y se comparte la caché de los servidores DNS a los que se le reenvían las consultas.

Lógicamente las consultas que se reenvían son sólo aquellas para las que el servidor no está autorizado, ni están previamente cacheadas.

- Servidor **sólo autorizado**. Son los servidores que son autorizados para una o varias zonas pero que no responden a peticiones que no sean para su zona. Esto implica que no preguntan a otros servidores, no hacen de reenviador, ni tampoco actúan como caché.

En Internet existen una serie de servidores DNS denominados servidores raíz (*root servers*) autorizados para el dominio raíz “.” Estos servidores contienen el archivo de la zona “.” en donde se delega a otros servidores DNS autorizados los dominios de primer nivel.

El ICANN es responsable de estos servidores raíz, en concreto 13, de los cuales existen copias repartidas mundialmente. Cada uno de ellos, y sus copias, se identifica con una misma dirección IP, de forma que cuando un cliente u otro servidor DNS les realiza una pregunta, responde la copia más cercana.

Como luego veremos, estos servidores raíz deben ser conocidos por todos los servidores DNS para poder responder a preguntas sobre nombres de dominio para los que no están autorizados.

En la web <http://root-servers.org/> podemos ver cuáles son los servidores raíz, sus nombres, ubicaciones y las empresas que los administran y en <http://www.iana.org/domains/root/db> los servidores de nombres y las empresas u organizaciones responsables de los dominios de primer nivel.

## 5. Clientes DNS (resolvedores).

Antes de que existieran servidores DNS que resolvieran los dominios, se usaba un archivo de texto (denominado *hosts*) donde se guardan las correspondencias entre nombres de dominio y direcciones IP. Este mecanismo se dejó de utilizar cuando Internet empezó a crecer en nombres de dominio, pasándose a usar servidores DNS.

Muchos sistemas operativos están configurados para usar este método antes de usar el servicio DNS. Así, si mediante la tabla de *hosts* no se puede resolver, se usa el servicio DNS. En una red local con pocos equipos y que mantengan direcciones fijas puede ser efectivo mantener la tabla de *hosts*, en otro caso, mantener actualizado y sincronizado el archivo *hosts* en todos los equipos es muy complicado y genera errores. En la actualidad también se usa para bloquear contenidos de Internet, como la publicidad web.

El formato del archivo *hosts* es el siguiente:

- En cada línea se debe introducir la dirección IP a la que resolverá, uno o más espacios o tabulaciones y el dominio a resolver.
- Se pueden introducir más de un dominio a resolver en la misma línea separados por uno o más espacios o tabulaciones.
- Cada correspondencia de dirección IP y dominio debe ir en una línea distinta.
- Las líneas que comienzan por # se consideran comentarios y no se computan.

Un ejemplo válido de archivo *hosts* podría ser este:

```
#Definición de localhost
127.0.0.1    localhost

#Correspondencia para equipos de la red local
192.168.210.1    pc1.aula.izv
192.168.210.222  ns1.aula.izv

#Correspondencia para una página web
209.85.229.104  www.google.es

#Dominios de Internet bloqueados mediante una IP inválida (como por ejemplo una máscara de subred...)
255.255.255.0  www.fantomas.com www.patapalo.com www.desplumadores.es
```

La única entrada obligatoria y que aparece siempre por defecto, es la del dispositivo de red loopback. Todas las direcciones del rango 127.0.0.0/8 son direcciones de loopback, de la cual la que se utiliza de forma mayoritaria es la 127.0.0.1 por ser la primera de dicho rango, pero podría utilizar cualquiera otra de dicho rango.

Esta dirección especial se suele utilizar cuando una transmisión de datos tiene como destino el propio equipo de forma que ésta se utiliza para dirigir el tráfico hacia ellos mismos. También es posible hacer *ping* a la dirección de loopback para probar la configuración de TCP/IP en el host local.

Un resolvedor (*resolver*) es cualquier software que realiza preguntas a un servidor DNS y entiende las respuestas recibidas. Los sistemas operativos incluyen un conjunto de librerías que realizan estas operaciones y que son invocadas por las aplicaciones cuando usan un nombre de dominio.

Normalmente, se puede configurar si el resolvedor deberá buscar primero en el archivo de *hosts* antes de hacer la resolución del nombre mediante el servicio DNS.



En general los pasos que sigue el resolver cuando una aplicación quiere resolver un nombre son:

1. Consulta la memoria caché de resolución de nombres del *hosts*, si está configurada.
2. Si la respuesta no es positiva, realiza la búsqueda en el archivo *hosts* del equipo local.
3. Si el nombre de dominio tampoco existe en el archivo *hosts*, se realizará una consulta *recursiva* al servidor DNS que se tenga configurado y éste suministrará la respuesta a la aplicación.

## 6. Mecanismo de resolución.

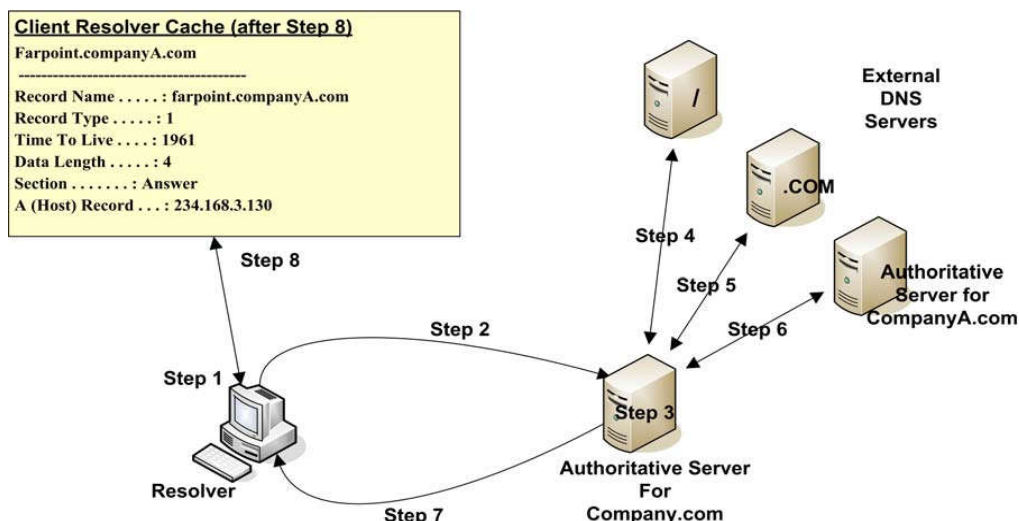
Las consultas a un servidor DNS pueden ser *recursivas* o bien *iterativas* (también llamadas no recursivas). Como veremos a continuación, a la hora de resolver un nombre, el *resolver* realiza una consulta recursiva, la cual podrá generar o no en una serie de consultas iterativas.

- Una consulta recursiva es aquella en la que el servidor siempre debe dar una respuesta completa o exacta. Esta respuesta podrá ser una de las tres: positiva (es decir, da la información) indicándose si es autorizada o no, negativa (si no se pudo resolver) o de error (por ejemplo por fallo en la red).
- Una consulta iterativa es aquella en la que el servidor DNS puede proporcionar una respuesta parcial. En este caso, a parte de las respuestas positivas, negativas o de error, puede dar una respuesta incompleta que indique una referencia a otros servidores a los que se les puede preguntar para resolver la pregunta.

En una consulta recursiva, el cliente realiza la pregunta a su servidor DNS, y éste tiene que darle una respuesta completa, realizando consultas a otros servidores DNS si fuera necesario para darle la respuesta. En una consulta iterativa, el servidor puede dar como respuesta información de a qué otros servidores DNS debería preguntar de nuevo el cliente para obtener respuesta a su pregunta. En este caso, el cliente es el encargado de ir haciendo preguntas a diferentes servidores para obtener una respuesta final.

Una consulta recursiva la inicia un cliente DNS a través del resolutor, o bien un servidor DNS que la traslada a otro servidor DNS actuando como reenviador. El proceso de resolución cuando un servidor recibe una consulta recursiva es el siguiente:

1. Si el servidor es autorizado para alguna zona, comprueba sus archivos de zona, y si encuentra la respuesta, responde indicando que la respuesta es *autoritativa*.
2. Si no encuentra la respuesta, o no es autorizado y actúa como caché, consulta su caché de respuesta anteriores, y si la encuentra responde que la respuesta es *no autoritativa*.
3. Si en el paso anterior no se encontró respuesta positiva, y tiene configurados reenviadores, entonces reenvía la consulta recursiva a este otro servidor DNS y la respuesta que reciba de éste la traslada al cliente o al servidor que le preguntó.
4. Si no tiene configurados reenviadores entonces:
  - a. El servidor inicia una serie de consultas *iterativas* a otros servidores DNS empezando la primera de ellas por un servidor raíz.
  - b. Los servidores consultados devuelven *referencias* a otros servidores DNS que se usan para realizarles la pregunta. Este proceso de preguntas a distintos servidores finaliza cuando finalmente un servidor autorizado proporciona una respuesta positiva o negativa.



La figura muestra los pasos cuando un cliente realiza una pregunta recursiva por el dominio “*farpoint.companyA.com*” a un servidor que tiene activada la recursividad (tiene que dar una respuesta completa). Como el servidor DNS no está autorizado para la zona “*companyA.com*”, no tendrá en sus archivos de zona el nombre de dominio por el que se le ha preguntado.

Como debe dar una respuesta, empezará una serie de consultas iterativas empezando por uno de los servidores raíz. El servidor raíz responde con una referencia al servidor autorizado para el dominio “*com*”. El servidor DNS envía una consulta iterativa al servidor autorizado del dominio “*com*”, y éste le responde con una referencia al servidor autorizado para el dominio “*companyA.com*”. Posteriormente el servidor DNS envía una consulta iterativa al servidor autorizado del dominio “*companyA.com*”.

El servidor autorizado del dominio “*companyA.com*” consulta en sus ficheros de zona y como existe el nombre de dominio “*farpoint.companyA.com*” responde con la información asociada a él, en este caso con la dirección IP (respuesta positiva). Si no existiera el nombre de dominio en sus archivos de zona, la respuesta sería negativa.

Finalmente el servidor que recibió la respuesta recursiva entrega al resolver la información por la que preguntó y la guarda en su caché disponible para futuras consultas.

Por lo tanto, las consultas iterativas son iniciadas por un servidor DNS a otro servidor DNS, cuando en una consulta recursiva no ha encontrado la respuesta en sus archivos de zona o en la caché.

En este proceso de resolución es muy importante la caché. Por ejemplo, si el resolutor pregunta primero por el dominio [www.granada.es](http://www.granada.es) y luego por [www.alomartes.es](http://www.alomartes.es) no tendrá que preguntar de nuevo a un servidor raíz, debido a ya que tendrá almacenada en la caché las direcciones de los servidores DNS autorizados para el dominio “*es*”.

Como hemos podido comprobar, las consultas recursivas son costosas para los servidores DNS ya que deben dar siempre una respuesta; de hecho los servidores DNS raíz y los de primer nivel (TLD) no responden nunca a consultas recursivas.

## 6.1. Respuestas en caché.

Las respuestas almacenadas en cache pueden ser positivas o negativas. Es decir se puede almacenar los registros de recursos de nombres de dominio resueltos (respuesta positiva), y la información de que no existe un registro de recursos para el nombre de dominio consultado (respuesta negativa). En este último caso se impide la repetición de preguntas para nombres que no existen.

Los archivos de zona almacenan los tiempos máximos que guardan las respuestas en caché (TTL), de forma que habrá TTL para las respuestas positivas y TTL para las negativas. Se recomienda que el TTL para las respuestas positivas sea mayor de un día, incluso semanas, y que el TTL para las respuestas negativas sea menor de 3 horas.

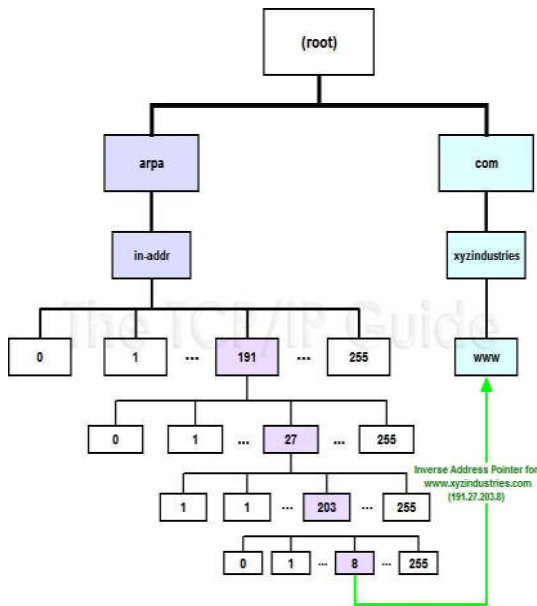
El administrador es el que tiene que valorar estos tiempos en función de la frecuencia con la que cambian los registros de recursos. Si el TTL para las preguntas positivas es pequeño, la información del dominio será más congruente pero a costa de aumentar el trabajo de los servidores y la carga de la red.

## 7. Correspondencias inversas.

Las resoluciones de nombre de dominio que hemos comentado hasta ahora se denominan de *resolución directa*. Las resoluciones inversas consisten en preguntar por una dirección IP para obtener el nombre o nombres de dominio como respuesta.

Uno de los principales motivos por el que se realiza una resolución inversa está ligado a la seguridad. Si al realizar una resolución directa para un nombre de dominio obtenemos una dirección IP, la cual se usa para realizar una resolución inversa, se debería obtener el nombre de dominio por el que preguntamos. Si no es así, posiblemente un intruso esté resolviendo nombres a direcciones IP no válidas con fines maliciosos. Los servidores DNS pueden configurarse para que actúen de esta forma para asegurarse de las respuestas recibidas.

También se suelen usar para detectar errores en la configuración de los servidores y equipos, *spam* en los servidores de correo, o para seguir la traza de un ataque.



En la resolución inversa, las direcciones IP se tratan como nombres donde cada byte de la IP es un dominio que cuelga del dominio `in-addr.arpa`. Así cuando realizamos una pregunta inversa del tipo ¿cuál es el nombre de dominio para la IP 192.168.210.1?, lo que realmente estamos preguntando es por el nombre de dominio de `1.210.168.192.in-addr.arpa`. Como vemos, la estructura jerárquica de la dirección IP tratada como nombre de dominio es de izquierda a derecha comenzando por el dominio `in-addr.arpa`.

Para la resolución inversa los servidores de nombres deben almacenar zonas de resolución inversa que podrán ser maestras o esclavas.

Las zonas directas e inversas son independientes, y el administrador debe mantener la información de ambas sin discrepancias. Si se administra una zona directa no es obligatorio administrar la zona inversa correspondiente. De hecho si administramos un dominio asociado a una IP pública contratada a un ISP (*Proveedor de Servicios Internet*), normalmente

el ISP no incluirá nuestro dominio en su zona inversa salvo que se lo pidamos expresamente.

El proceso de resolución inversa es análogo al directo. Por ejemplo una consulta recursiva de la IP 191.27.203.8 a un servidor DNS, empezaría en buscar en la caché, y si no tiene la respuesta comenzaría una serie de consultas iterativas a los servidores DNS raíz, después a los servidores autorizados para el dominio `191.in-addr.arpa`, luego a los autorizados para el dominio `27.191.in-addr.arpa`, hasta realizar la consulta al servidor autorizado para el dominio `203.27.191.in-addr.arpa`, el cual debe dar una respuesta completa.

Los archivos de zona de resolución inversa usan registros de recursos de tipo NS y PTR. Por ejemplo para el siguiente archivo de zona de resolución directa para el dominio `aula.izv`:

<code>aula.izv.</code>	IN	NS	<code>puesto1.aula.izv.</code>
<code>puesto1.aula.izv.</code>	IN	A	<code>192.168.210.1</code>
<code>puesto2.aula.izv.</code>	IN	A	<code>192.168.210.2</code>
<code>puesto3.aula.izv.</code>	IN	A	<code>192.168.210.3</code>
<code>www.aula.izv.</code>	IN	CNAME	<code>puesto2.aula.izv.</code>

tendríamos el siguiente archivo de zona de resolución inversa `210.168.192.in-addr.arpa`. Este archivo permitiría resolver las consultas inversas sobre direcciones IP de la red `192.168.210.0/24`.

<code>210.168.192.in-addr.arpa.</code>	IN	NS	<code>puesto1.aula.izv.</code>
<code>1.210.168.192.in-addr.arpa.</code>	IN	PTR	<code>puesto1.aula.izv.</code>
<code>2.210.168.192.in-addr.arpa.</code>	IN	PTR	<code>puesto2.aula.izv.</code>
<code>3.210.168.192.in-addr.arpa.</code>	IN	PTR	<code>puesto3.aula.izv.</code>

## 8. Registros de recursos.

Los archivos de zona almacenan la información sobre nombres de dominio mediante *registros de recursos* (RR, Resource Records). Estos RR son los que se envían entre los clientes y servidores DNS en las preguntas y respuestas.

Los RR se representan en los archivos de zona en formato texto, y de forma binaria en los mensajes que se envían mediante el protocolo DNS. El formato de los RR en modo texto es:

Nombre de Dominio	[TTL]	Clase	Tipo	Dato
-------------------	-------	-------	------	------

Por ejemplo:

<code>puesto1.aula.izv.</code>	3600	IN	A	<code>192.168.210.1</code>
--------------------------------	------	----	---	----------------------------

Lo primero es el nombre de dominio que se asocia al recurso, opcionalmente aparece el tiempo de vida medio (TTL) que indica en segundos el tiempo que puede estar el registro en caché antes de ser descartado. (Veremos que se puede especificar un tiempo global para todos los registros de la zona).

La clase define el tipo de protocolo usado. Aunque se pueden usar otras arquitecturas la habitual es Internet (IN). A continuación el tipo indica el tipo de dato asociado al nombre de dominio y por último se especifica el dato asociado al nombre de dominio.

## 8.1. Tipos de registros.

Comentaremos sólo los tipos de registros de recursos más usados, ya que el protocolo define alrededor de 30 tipos diferentes. En el sitio web del IANA se pueden consultar todos ellos.

- Registro SOA (Star of Authority). Es el primer registro de una zona y permite definir el tipo de servidor y las opciones generales de la zona.

El registro tipo SOA comienza con el nombre del dominio de la zona, seguido opcionalmente de un TTL (lo normal es usarlo para así establecer de forma global el TTL para toda la zona), continua con la clase IN, seguido del tipo SOA y finaliza con los datos asociados. Estos datos asociados son:

- FQDN del servidor de nombres maestro para el dominio.
- Contacto. Dirección de correo del responsable del dominio. (La arroba se sustituye por “.”)
- Número de serie (*serial*). Indica la versión del archivo de zona que se irá incrementando cada vez que el archivo se modifique. De esta forma los servidores secundarios conocen si el archivo de zona ha cambiado y deben actualizarse mediante una transferencia de zona.
- Actualización (*refresh*). Indica cada cuanto tiempo deben contactar los servidores secundarios con el servidor maestro para comprobar si ha habido cambios en la zona. Dependiendo de la frecuencia de actualización de la zona primaria se usará más o menos tiempo. Se recomiendan valores entre 20 minutos y 212 horas. (Si se usan notificaciones, se puede usar un valor alto).
- Reintentos (*retry*). Indica cada cuanto tiempo deben reintentar los servidores secundarios contactar con el servidor maestro si éste no responde a una actualización.
- Caducidad (*expire*). Tiempo durante el cual un servidor secundario puede estar sin contactar con el primario para comprobar la zona. Si se supera este tiempo, el secundario descarta los datos que tenía sobre la zona y se declara no autorizado para la zona. (Se recomienda entre 2 y 4 semanas)
- TTL negativo. Es el tiempo mínimo en que se almacena las respuestas negativas sobre la zona.

Por ejemplo, nuestro dominio aula.izv podría tener el siguiente registro SOA

```
aula.izv. IN SOA puesto1.aula.izv. remaster.aula.izv. (
    2019102301      ; nº de versión del archivo de zona
    604800          ; tiempo de refresco
    86400           ; tiempo de reintento
    2419200         ; tiempo de expiración
    38400 )         ; TTL negativo
```

- Registro NS (Name Server). Permite indicar los servidores de nombres autorizados para una zona. (Cada zona debe contener al menos un registro NS, por ejemplo un maestro; y puede tener uno o más registros NS para indicar los esclavos). Este registro también permite indicar los nombres de los servidores con autoridad de los subdominios que hayan sido delegados. (Ver ejemplo de la página 6).
- Registro A. (Address). Establece una correspondencia entre un nombre de dominio FDQN y una dirección IPv4.
- Registro AAAA. (Address). Establece una correspondencia entre un nombre de dominio FDQN y una dirección IPv6.
- Registro CNAME (Canonical Name). Indica un alias o sobrenombre para los nombres de dominio previamente especificados en registros A y AAAA. Un alias puede apuntar a otro dominio, pero no se pueden usar en la parte derecha de registros MX y NS, ya que estos dos tipos de registros necesitan siempre usar en su parte derecha nombres que aparezcan en registros de tipo A o AAAA.
- Registro MX (Mail Exchange). Define los equipos encargados del correo en el dominio, para que los agentes de transporte de correo sepan a qué equipo deben entregar los mensajes. Se pueden definir varios servidores de correo donde en el RR de cada uno de ellos se indica mediante un número el orden de preferencia de cada uno de ellos; a número menor le corresponde mayor preferencia.

- Registro SRV (Services Record). Permite definir equipos que actúan como servidores de algún servicio particular en el dominio. Por ejemplo servidores LDAP, servidores de mensajería, etc.
- Registro PTR (Pointer Record). Establecen correspondencias inversas, es decir, entre direcciones IP y nombres de dominio. Por lo tanto sólo se usan en las zonas de resolución inversas. Si se usan direcciones tipo IPv4 e IPv6 deben aparecer en zonas separadas.
- Registro TXT (Texto). Permite registrar cualquier texto que tenga relación con un equipo.

## 8.2. Registros pegamento (Glue Record).

Sabemos que un servidor de nombres se puede configurar para delegar o no algunos de sus subdominios en otros servidores de nombres. Podemos encontrar dos situaciones.

En el caso de que el servidor de nombres autorizado para el subdominio delegado se encuentre en el mismo dominio es necesario añadir:

1. Un registro NS en el dominio que delega, para indicar cuál es el servidor de nombres para la zona delegada.
2. Un registro A que indique la IP del servidor de nombres autorizado del subdominio delegado. Este tipo de registro se denomina *glue record* porque que de alguna forma relaciona o pega la zona hija con la zona padre.
3. OJO. Si el dominio que delega usa reenviadores, cuando le consultemos por un nombre del subdominio delegado, nuestro servidor DNS detectará que no tiene autoridad sobre él, y por lo tanto le preguntará al servidor DNS indicado en el *forwarders* sin realizar la delegación.

Si usamos bind9, para solucionar este problema tendríamos que anular la función de *reenvío* pero sólo para la zona que se delega, y para ello en el fichero */etc/bind/named.conf.local*, donde se definen las zonas DNS, configuraríamos en la zona que delega el parámetro: `forwarders { };`

Por lo tanto, los servidores de un dominio deben conocer la dirección IP de los servidores de nombres de los subdominios, para que los clientes puedan dirigirse a ellos en las consultas. (Ver ejemplo pag. 6).

En el caso de que el servidor de nombres autorizado para el subdominio delegado no se encuentre en el mismo dominio, sólo es necesario añadir el registro NS que indique el servidor de nombres de la zona delegada. En este caso no necesita un registro tipo A, ya que los clientes podrán resolver el nombre de dominio de la zona delegada normalmente.

Finalizamos con un ejemplo de zona de resolución directa para el dominio *aula.izv.* con un subdominio delegado en un subdominio propio y otro delegado en un subdominio ajeno.

```

aula.izv. IN SOA puesto1.aula.izv. carlos.aula.izv. (
                2012041401      ; nº de versión del archivo de zona
                604800           ; tiempo de refresco
                86400            ; tiempo de reintento
                2419200          ; tiempo de expiración
                38400 )          ; TTL negativo
aula.izv.      IN      NS      puesto1.aula.izv.
puesto1.aula.izv. IN    A      192.168.210.1
puesto2.aula.izv. IN    A      192.168.210.2
puesto3.aula.izv. IN    A      192.168.210.3
www.aula.izv.  IN      CNAME   puesto2.aula.izv.
ftp.aula.izv.  IN      CNAME   puesto2.aula.izv.
aula.izv.      IN      MX      10      puesto3.aula.izv.
smtp.aula.izv. IN      CNAME   puesto3.aula.izv.
_ldap._tcp.aula.izv. IN    SRV   0 0 389 puesto3.aula.izv.

;subdominio sri.aula.izv. delegado en un subdominio propio
sri.aula.izv.  IN      NS      puesto10.sri.aula.izv.
puesto10.sri.aula.izv. IN    A      192.168.210.10      ; (Glue Record)

;subdominio bd.aula.izv. delegado en un subdominio ajeno
bd.aula.izv.  IN      NS      ns1.informatica.izv.

```

## 9. Transferencias de zona.

Los servidores en los que se declaran zonas esclavas, deben obtener los archivos de zonas, es decir los registros de recursos, de otros servidores (maestros o esclavos) autorizados para dichas zonas.

Este proceso se denomina transferencia de zona y hay que configurar los servidores para que las realicen. Los servidores maestros usan el puerto 53/TCP para realizar el intercambio de zona la cual puede ser de dos tipos: *completa* e *incremental*.

En las transferencias de zonas completas, el servidor maestro envía al esclavo todos los datos de la zona sustituyendo a los datos anteriores. En las transferencias de tipo incremental, el maestro sólo envía los datos que han cambiado desde la última transferencia de zona.

Las transferencias de zona puede comenzar por una pregunta del servidor esclavo al maestro para comprobar si hay algún cambio en la zona. La primera vez que se inicia el servidor esclavo realiza esta pregunta, y luego la repite cada cierto tiempo (especificado en el campo *refresh* del registro SOA).

Por otro lado el servidor maestro puede ser el encargado de notificar a los servidores esclavos de las modificaciones producidas en sus zonas. De esta manera, si se produce una modificación en la zona del maestro, el esclavo queda enterado de forma inmediata y puede comenzar el proceso de transferencia.

En el proceso de transferencia, el maestro envía su registro SOA al esclavo, de forma que éste obtiene su número de serie y lo compara con el que tiene almacenado en su zona. Si el número de serie que obtiene es superior al suyo, entiende que sus datos no están actualizados y se realiza la transferencia.

El que la transferencia sea de tipo completa o incremental depende del tipo de petición que envíe el servidor esclavo al maestro, empleando para ello mensajes de tipo AXFR o IXFR respectivamente.

## 10. DNS dinámico (DDNS, Dynamic DNS)

Hasta ahora hemos comentado que es responsabilidad del administrador mantener actualizada la zona de los servidores de nombres. Esta tarea puede ser muy costosa si la organización dispone de varios dominios o múltiples servidores DNS. Si además si se usan servidores DHCP para asignar direcciones a los equipos, el mantenimiento manual de las zonas no es una opción viable. Además cada vez que se realizara una modificación, se tendría que parar y reiniciar el servicio DNS.

Para aliviar esta situación, el RFC 2162 define los procesos para que de forma automática, los registros de recursos de una zona se puedan actualizar de forma externa, sin que el administrador tenga que editar manualmente los archivos de zona y sin necesidad de reiniciar el servicio DNS.

Normalmente las actualizaciones dinámicas de los registros de una zona las pueden realizar los propios equipos clientes o bien los servidores DHCP.

En el primer caso, hay que configurar los clientes de cada equipo y el servidor DNS para que permita las actualizaciones por parte de los equipos. Si las realizan los servidores DHCP, se deberá configurar el servidor DHCP de manera que si se permite el cambio de los nombre de dominio, podrán ser los clientes DHCP los que se configuren para que envíen su nombre al servidor DHCP o bien sea el propio servidor DHCP el que le asigne el nombre al equipo; también hay que configurar el servidor DNS para que permita las actualizaciones por parte del servidor DHCP.

Para controlar quién puede realizar estas actualizaciones, se suelen usar ACL (*listas de control de acceso*) para determinar desde que IPs se pueden realizar las actualizaciones, acompañado de algún protocolo de autenticación simple como es TSIG (*Transaction SIGnature*) o TKEY (*Transaction Key*) para aumentar la seguridad.

Otro problema para los usuarios que usan el servicio DNS para acceso a Internet, es cuando quieren acceder a un servicio (servidor web, ftp, correo, etc.) en un dominio que no tiene una IP fija en Internet.

Actualmente, la mayoría de los ISP proporcionan direcciones IP públicas por DHCP al *router* que nos conecta a Internet. Esto implica que la IP cambiará cada cierto tiempo y por lo tanto no podemos asignarle un nombre de dominio estable a esta dirección IP.

Los llamados DNS dinámicos que ofrecen algunos sitios web en Internet permiten registrar un nombre de dominio actualizándolo con la dirección IP que realmente se tenga en el momento. Lo más habitual es que sea el propio *router* el que tenga la posibilidad de actualizar el servidor DNS cuando cambie su dirección IP, aunque también podría ser un programa instalado en algún equipo de la red.

Varias web, como DynDNS, No-ip, EasyDNS, DonDNS, etc. ofrecen estos servicios de forma gratuita sobre un dominio de segundo nivel de su propiedad. Por ejemplo podríamos configurar un servidor web con IP dinámica con el nombre de dominio “micasa.dyndns.org”.

## 11. Seguridad DNS.

El servicio DNS es fundamental para el funcionamiento de una red e imprescindible en Internet, por ello suele ser un objetivo para los atacantes. El hecho de que el protocolo DNS se diseñara inicialmente sin tener en cuenta la seguridad (al igual que otros servicios de red) y que el servicio, al ser distribuido dependa de muchos equipos, implica ciertas dificultades para su administración.

Las principales amenazas para un servicio DNS son:

- Ataques al servidor DNS usando *exploits* aprovechando agujeros en la seguridad.
- Modificaciones de los archivos de zonas aprovechando una mala configuración de los permisos de usuarios que puedan acceder al equipo servidor de forma remota.
- Ataques DoS (*Denial of Service*) mediante inundación de múltiples peticiones de resolución.
- Suplantación del servidor DNS, enviando RR incorrectos y envenenamiento de caché de clientes.
- Envenenamiento de la caché de un servidor DNS al preguntar a otro DNS suplantado.
- Suplantaciones del servidor maestro en las transferencias de zona.
- Suplantaciones de los orígenes externos que envían actualizaciones a los DNS dinámicos.

Los mecanismos de seguridad que se pueden establecer son:

1. En los servidores DNS. Actualización a las últimas versiones software, instalación de parches de seguridad, enjaular el servidor en un entorno seguro, configuración de los privilegios de acceso a los archivos de zona, distribuir los servidores DNS en distintas redes y ubicaciones.
2. En las transferencias de zona y actualizaciones dinámicas.
  - Establecer ACL (*listas de control de acceso* por IP o nombres de dominios) con los servidores desde los que se permiten las transferencias de zona y las actualizaciones.
  - Utilizar cortafuegos para controlar las transferencias y actualizaciones de zona.
  - Mecanismos de autenticación de servidores y equipos mediante generación de claves del tipo TSIG o TKEY.
3. En las consultas DNS.
  - Configurar los servidores para minimizar las consultas recursivas.
  - Limitar por dirección IP los equipos que pueden preguntar a un servidor.
  - Implantación de DNSSEC (DNS Security) basado en algoritmos de cifrado asimétricos para garantizar la autenticidad e integridad en las consultas y respuestas DNS.

## 12. Servidores DNS en Linux.

El servidor DNS más extendido en sistemas operativos basados en UNIX, y de facto un estándar, se llama **bind** (*Berkeley Internet Name Domain*), y actualmente lo desarrolla y mantiene la organización sin ánimo de lucro ISC (Internet Systems Consortium).

Antes de realizar la instalación, necesitaremos siempre poder resolver el nombre de nuestro servidor independientemente de que tengamos un servidor DNS. Al inicio del sistema, puede suceder que sea necesario resolver el nombre del equipo, pero que no se logre debido a que todavía no esté operativa la red o el servicio DNS, o que no podamos acceder al servidor DNS. Para ello, se comprueba que el nombre del servidor esté correctamente establecido en los ficheros `/etc/hostname` y `/etc/hosts`.

Por ejemplo, el contenido de un fichero `/etc/hostname` podría ser:

```
ubup
```

Y el contenido del fichero `/etc/hosts` suponiendo que el equipo pertenece al dominio `aula.izv`:

```
127.0.0.1    localhost
127.0.1.1    ubup.aula.izv ubup
::1         ip6-localhost ip6-loopback
```

También, como suele suceder en la mayoría de los servicios, el servidor deberá tener una IP fija.

El servidor **bind** se instala mediante el paquete *bind9*, sus archivos de configuración se ubican en el directorio */etc/bind*, y el demonio del proceso servidor se denomina *named*.

Dependiendo de la distribución, los scripts que permiten arrancar, parar y reiniciar el servicio son:

- */etc/init.d/bind9 {start|stop|reload|restart|force-reload|status}*
- *service bind9 {start|stop|reload|restart|force-reload|status}*
- *systemctl {start|stop|reload|restart|force-reload|status} bind9.service*

Terminada la instalación debemos comprobar que el proceso *named* está en ejecución, y que el servicio está a la escucha en los puertos 53/TCP y 53/UDP. Para ello ejecutamos:

```
$ ps -ef | grep named
$ netstat -ltun
```

Cuando se instala **bind**, se generan unos archivos con una configuración básica, con unas zonas ya preconfiguradas:

El archivo */etc/bind/named.conf* es el archivo de configuración principal y mediante la directiva “include”, se incluyen las configuraciones de los tres archivos siguientes:

- */etc/bind/named.conf.options*. Contiene las opciones de configuración generales del servidor y define su comportamiento de forma global.
- */etc/bind/named.conf.local*. Es el archivo de configuración de zonas y donde se declaran las zonas de resolución tanto directas como inversas. Inicialmente vacío.
- */etc/bind/named.conf.default-zones*. Contiene la definición de las zonas creadas por defecto. Los archivos de las zonas por defecto, y referenciados desde */etc/bind/named.conf.default-zones* son:
  - */etc/bind/db.root* Servidores raíz.
  - */etc/bind/db.local* Resolución directa del bucle local. (Resuelve localhost)
  - */etc/bind/db.127* Resolución inversa del bucle local. (Resuelve 127.0.0.1)
  - */etc/bind/db.0* Resolución directa *broadcast*.
  - */etc/bind/db.255* Resolución inversa *broadcast*.

Siempre que se cambie la configuración del servidor hay que reiniciar el servicio para que tengan efecto los cambios. Es muy conveniente examinar el archivo de *logs* (*/var/log/syslog*) del sistema para comprobar que no se hayan producido fallos en el arranque del servidor.

Como el contenido del archivo de configuración */etc/bind/named.conf.options* que se genera por defecto es muy simple. Le vamos a añadir algunas opciones más cuyo significado aparecen como comentarios.

```
options {
    // Directorio de trabajo por defecto
    directory "/var/cache/bind";

    // Para aceptar sólo peticiones de resolución de ciertos equipos.
    allow-query { 127.0.0.1; 192.168.210.0/24; };

    // Por defecto se permiten consultas recursivas (recursion yes;) pero podemos limitar dichas consultas
    // recursivas solo a los equipos que indiquemos
    allow-recursion {127.0.0.1; 192.168.210.0/24; };

    // En principio no se permite enviar notificaciones de actualización de zona a otros servidores
    notify no;

    // Servidores DNS a los que reenviar peticiones de información de otras zonas
    // forwarders {80.58.61.250; };

    // No se responderá autoritativamente a peticiones de nombre de dominio inexistentes
    auth-nxdomain no;    # conform to RFC1035

    // Direcciones y puertos en los que realizar la escucha de peticiones de resolución.
    // (Por si hay varias interfaces de red...)
    listen-on port 53 { 127.0.0.1; 192.168.210.211; };
    // también escucha en Ipv6 (y además por cualquier interfaz de red...)
    listen-on-v6 { any; };

    // no se usa validación entre clientes y servidores usando firmas digitales
    dnssec-validation no;
};
```



Existen muchas más opciones disponibles, algunas de éstas irán apareciendo posteriormente.

#### - Configuración del servidor DNS como solo caché.

Por defecto, el servidor **bind** está configurado como *solo caché* ya que no está autorizado para ninguna zona y es capaz de responder a consultas recursivas (por defecto el parámetro *recursión* es *yes*).

Podemos probarlo configurando un cliente con dicho servidor DNS y preguntar por cualquier nombre de dominio en Internet. Además las consultas repetidas se aceleran al estar resueltas en la caché.

#### - Configuración del servidor DNS como reenviador (*forwarding*).

Para configurar un servidor para que reenvíe las consultas a otros servidores (*forwarding*) se edita el archivo `named.conf.options` y mediante el atributo *forwarders* se indican las direcciones IP separadas por punto y coma de los DNS que actuarán como reenviadores (*forwarders*).

Recordemos que las consultas que se reenvían son sólo aquellas para las que el servidor no está autorizado y sus respuestas no están en caché.

Si el servidor está configurado como reenviador, cabe la posibilidad de usar la opción *forward* la cual ya tiene como valor por defecto el valor *first*. Con *first* el servidor consultará a los servidores indicados con *forwarders* en primer lugar, y si no obtiene la respuesta, entonces intentará buscarla por sí mismo. Si se usa *forward only*, entonces el servidor sólo consultará a los reenviadores.

#### - Configuración del servidor DNS como maestro para una zona de resolución directa y una zona de resolución inversa.

Para configurar el servidor DNS como maestro para una zona de resolución directa y una zona de resolución inversa se debe editar el archivo `named.conf.local`. Aquí se indican entre otras opciones el nombre de la zona, el tipo de zona y el archivo donde se almacenarán los registros de recursos de la zona.

Por ejemplo, si queremos que el servidor tenga autoridad para el dominio `aula.izv` y para la zona de resolución inversa de la red `192.168.210.0/24` editaríamos el archivo `named.conf.local` y le añadimos las zonas directa e inversa:

```
zone "aula.izv" {
    type master;
    file "/etc/bind/db.aula.izv";
};
zone "210.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168.210";
};
```

A continuación deberíamos crear los archivos con los registros de recursos de ambas zonas. Los archivos a crear son los que hemos indicado anteriormente en el parámetro *file*. Si no hubiéramos especificado la ruta absoluta, se tomaría el directorio de trabajo por defecto (`/var/cache/bind`).

Crear los archivos con los registros de recursos editándolos directamente suele causar problemas, ya que la sintaxis es algo complicada. Lo más cómodo es usar alguna herramienta gráfica que nos ayude en esta tarea como es “webmin”. Además podemos indicarle que a medida que vayamos registrando los registros tipo A de la zona directa, cree de forma automática los correspondientes PTR de la zona inversa.

Ejemplo de archivo de zona de resolución directa creado con “webmin”:

```
$ttl 38400 ;
aula.izv. IN SOA ubup.aula.izv. carlos.aula.izv. (
                2016101401      ; nº de versión del archivo de zona
                604800          ; tiempo de refresco
                86400           ; tiempo de reintento
                2419200          ; tiempo de expiración
                38400 )          ; TTL negativo
aula.izv. IN NS ubup.aula.izv. ; servidor maestro autorizado
ubup.aula.izv. IN A 192.168.210.211 ; correspondencia FDQN a IP
winp.aula.izv. IN A 192.168.210.212
profe.aula.izv. IN A 192.168.210.210
```

smtp.aula.izv.	IN	CNAME	profe.aula.izv.	; alias para nombres de dominio
www.aula.izv.	IN	CNAME	profe.aula.izv.	
aula.izv.	IN	MX 10	profe.aula.izv.	; quien entrega el correo al dominio

---

Ejemplo de archivo de zona de resolución inversa

```
$ttl 38400 ;
210.168.192.in-addr.arpa. IN      SOA      ubup.aula.izv. carlos.aula.izv. (
                                1316802825
                                10800
                                3600
                                604800
                                38400 )
210.168.192.in-addr.arpa.      IN      NS      ubup.aula.izv.
211.210.168.192.in-addr.arpa.  IN      PTR     ubup.aula.izv.
212.210.168.192.in-addr.arpa.  IN      PTR     winp.aula.izv.
210.210.168.192.in-addr.arpa.  IN      PTR     profe.aula.izv.
```

En los archivos de zona puede usarse expresiones del tipo \$TTL 86400. Esto permite especificar un valor de TTL por defecto para todos los registros de recursos. Se puede usar el carácter “@” como sustituto del nombre de la zona definido en `named.conf.local` para no tener que repetirlo en el archivo.

Después de reiniciar el servicio y comprobar que no hay errores, deberíamos configurar los clientes para que usen como servidor DNS la máquina en donde hemos instalado y configurado el servidor.

#### - Configuración del servidor DNS como esclavo para una zona de resolución directa y una zona de resolución inversa.

En el servidor que se configure como esclavo para una zona, debe editarse el archivo `/etc/named.conf.local` e indicar el nombre de la zona (*zone*), el tipo de servidor (*type slave*), cual es el servidor maestro del que recibirá la transferencia (*masters*), y el archivo donde se almacenarán los registros de recursos que se reciban (*file*).

Por ejemplo, si el equipo donde se va a configurar un servidor esclavo para una zona de resolución directa fuera `profe.aula.izv` con IP `192.168.210.210`, escribiríamos en el archivo `/etc/bind/named.conf.local` de este servidor esclavo:

```
zone "aula.izv" {
    // tipo de servidor
    type slave;
    // IP del servidor maestro
    masters { 192.168.210.211; };
    file "/etc/bind/db.aula.izv";
};
```

En el archivo `/etc/named.conf.local` del servidor maestro de la zona, deberemos indicar que se permitan transferencias de zona hacia el servidor esclavo (*allow-transfer*), y además se puede configurar para que notifique a los servidores esclavos los cambios que se produzcan en el maestro (*notify*).

```
zone "aula.izv" {
    type master;
    file "/etc/bind/db.aula.izv";
    // servidores a los que se puede transferir la zona
    allow-transfer { 192.168.210.210; };
    // se notificará a los servidores esclavos de los cambios
    notify yes;
};
```

Por último, en el archivo de zona del servidor maestro, se debe indicar la existencia de otro servidor DNS para la zona, añadiendo para ello un nuevo registro NS con el nombre de dominio del esclavo:

```
...
aula.izv.      IN      NS      profe.aula.izv. ; nombre de dominio del servidor esclavo
profe.aula.izv. IN      A      192.168.210.210 ; resolución del nombre del servidor esclavo
```

Las transferencias de zona se pueden realizar usando el comando `dig` indicando como parámetros: el nombre de dominio o IP del servidor maestro de la zona, el nombre de dominio de la zona a transferir, y el tipo de transferencia a realizar (`axfr` para completa). Para nuestro ejemplo sería:

```
dig @192.168.210.211 aula.izv axfr
```

### - Configuración del servidor DNS para actualizaciones dinámicas (Dynamic DNS).

Si usamos el servidor DHCP del ISC, podemos configurar el servicio DNS *bind* para que admita actualizaciones dinámicas de forma relativamente fácil. Para la configuración, supondremos que ambos servidores corren en el mismo equipo (`ubup.aula.izv`).

En muchas instalaciones es frecuente que en el servidor DNS se creen de forma manual los registros de recursos correspondientes a los equipos que van a ofrecer un servicio (servidor web, ftp, correo, etc.); y que se configuren los servicios DNS y DHCP para que los registros de recursos de los equipos clientes se generen o actualicen de forma automática.

A continuación detallaremos los pasos necesarios a realizar para dicha configuración.

1) En primer lugar está el tema de la seguridad. Normalmente, aparte de indicar qué equipos podrán actualizar los archivos de zona del servidor DNS, se usa TSIG (Transaction **S**ignature) como método para firmar las transacciones y mensajes de DNS mediante el uso de claves simétricas compartidas, de forma que sólo estarán permitidas las actualizaciones que presenten la clave compartida autorizada.

En nuestro caso, la clave del servidor DNS y la de servidor DHCP que le enviará los datos de las actualizaciones deberá ser la misma (claves simétricas compartidas). Para generar la clave se usa el comando *dnssec-keygen* de la forma:

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST ddnsclave
```

En este caso, el comando genera una clave de 128 bits empleando un algoritmo de código de autenticación de mensaje sobre una función hash de tipo MD5. El parámetro `-n HOST` crea claves que permiten identificar equipos. Se puede aumentar la seguridad usando SHA256 como función hash.

El comando crea dos archivos: `Kddnsclave.+157+59571.key` y `Kddnsclave.+157+59571.private`; el segundo es el que nos interesa ya que incluye la clave secreta que necesitamos. Los números 157 en el nombre del archivo indican el tipo de algoritmo usado, y los 5 dígitos posteriores indican un identificador aleatorio.

Estos dos archivos deberán colocarse en el directorio donde el servidor DNS almacene los archivos de zona; en nuestro caso en `/etc/bind`.

2) El segundo paso sería configurar el servidor DNS para que acepte actualizaciones firmadas con esta clave. Para ello editamos el archivo `/etc/bind/named.conf.local` y le añadimos una entrada *key* con la definición de la clave.

Además deberemos configurar las zonas que recibirán las actualizaciones con dicha clave, usando la opción *allow-update*; o bien podemos afinar un poco más y especificar ciertas reglas mediante el uso de *update-policy*. El archivo quedaría de la forma:

```
key ddnsclave {
    algorithm hmac-md5;
    // la clave secreta se obtiene del archivo Kddnsclave.+157+59571.key
    secret "4vi9ZnoZlJ/axrD3ODwEbw==";
};
zone "aula.izv" {
    type master;
    file "/var/lib/bind/aula.izv.hosts";
    // permitimos que la clave "ddnsclave" se pueda usar para actualizar la zona
    allow-update { key ddnsclave; };
    // O bien indicamos reglas para la actualización
    // update-policy {grant ddnsclave wildcard *. aula.izv. A TXT; };
};
zone "210.168.192.in-addr.arpa" {
    type master;
    file "/var/lib/bind/192.168.210.rev";
    // permitimos que la clave "ddnsclave" se pueda usar para actualizar la zona
    allow-update { key ddnsclave; };
    // O bien indicamos reglas para la actualización
    // update-policy {grant ddnsclave wildcard *.210.168.192.in-addr-arpa. PTR; };
};
```

3) Ahora se configura el servidor DHCP para que actualice las zonas del servidor DNS cada vez que otorgue una concesión. Usaremos la configuración de ejemplo para el servidor DHCP del tema anterior y le añadiremos varias entradas. Para ello editamos el archivo de configuración `/etc/dhcp/dhcpd.conf`

```
# Configuración global
option domain-name-servers 192.168.210.211, 8.8.8.8; # DNS para los clientes
option domain-name "aula.izv."; # Nombre de dominio para los clientes
option subnet-mask 255.255.255.0; # Máscara por defecto para los clientes
default-lease-time 600; # Tiempo en segundos de la concesión
max-lease-time 7200; # Máximo tiempo en segundos que durará la concesión

# Activamos las actualizaciones DNS dinámicas
ddns-updates on;
# Indica el modo de actualización
ddns-update-style interim;
# los clientes DHCP no podrán realizar directamente las actualizaciones
ignore client-updates;

# indicamos la clave secreta (la misma que indicamos en el servidor DNS ya que es simétrica)
key ddnsclave {
    algorithm hmac-md5;
    secret "4vi9ZnoZIJ/axrD30DwEbw==";
}

# Establecemos las zonas a actualizar indicando dónde se halla el servidor DNS (su IP), y la clave a usar
zone aula.izv. { primary 127.0.0.1; key ddnsclave; }
zone 210.168.192.in-addr.arpa. { primary 127.0.0.1; key ddnsclave; }

# Especificación de un rango
subnet 192.168.210.0 netmask 255.255.255.0 {
    range 192.168.210.60 192.168.210.80; # Rango de IP desde la 60 a la 80 inclusive
    option broadcast-address 192.168.210.255; # Dirección de difusión para el rango
    option routers 192.168.210.254; # Puerta de enlace
    option domain-name-servers 192.168.210.211; # servidores DNS
    default-lease-time 3600; # Tiempo en segundos que durará la concesión

    # Si tuviéramos varias subnet en donde el servidor DHCP tuviera que actualizar varios dominios
    # se indicarían aquí, en cada subnet, en lugar de indicarse en la configuración general del servidor DHCP
    # ddns-domainname "aula.izv.;"
    # ddns-rev-domainname "in-addr.arpa.;"

    # Configuración particular para un equipo
    host puesto7 {
        hardware ethernet 00:0c:29:1e:88:1d;
        fixed-address 192.168.1.66;
        option host-name "ubu7";
    }
}
```

4) Terminada la configuración de los servidores DNS y DHCP, reiniciamos ambos servicios comprobando que no hay errores en los *logs* (`/var/log/syslog`). A partir de ahora, ya no podremos editar los archivos de zona manualmente para después reiniciar el servicio DNS. Una vez configurado el DNS dinámico, *bind* es quien se hace cargo de la actualización de los archivos de zona.

A partir de ahora, para modificar los archivos de zona tendremos que usar el comando *nsupdate* o bien aprovechar que cuando se para el servicio DNS, *bind* sobrescribe los archivos de zona con los datos actuales y con todas las actualizaciones. Por lo tanto si queremos modificar el archivo de la zona, pararemos *bind*, borramos los archivos *journal* (archivos con las actualizaciones), editamos manualmente los archivos de zona e iniciamos *bind*.

Los archivos *journal* (o de diario), son archivos binarios y no se pueden editar. Los nombres de dichos archivos coinciden con los nombre de archivo de la definición de las zonas añadiéndoles la extensión *.jnl*

5) Por último queda comprobar la configuración iniciando algún cliente configurado para obtener su configuración de red mediante DHCP. Dicho cliente NO puede estar registrado previamente de forma manual en los archivos de zona.

Los clientes Windows de forma general envían al servidor DHCP el nombre de equipo, lo que permitirá que DHCP envíe dicho nombre de dominio para actualizar las zonas del servidor DNS. En los

clientes Linux debemos asegurarnos que también lo hacen. Para ello deberemos editar el archivo de configuración del cliente `/etc/dhcpd/dhclient.conf` y comprobar que está activa la opción

```
send host-name "<hostname>";
```

Una vez iniciado el cliente, deberemos observar el archivo `/var/log/syslog` del equipo servidor para comprobar que la concesión otorgada por el servidor DHCP ha sido correcta y que los registros de zona del servidor DNS han quedado actualizados.

#### - Chequeo del servidor DNS y de zonas.

El servidor BIND dispone de dos herramientas para chequear la sintaxis de los archivos de configuración y para los archivos de la zona.

El comando `named-checkconf [archivo]` chequea la sintaxis de archivos de configuración del demonio `named` mostrando los errores en la salida. Si no se indica ningún archivo, se chequea por defecto del archivo de configuración principal `named.conf`. Si no genera una salida, significa que es correcto.

El comando `named-checkzone nombre-de-zona archivo-de-zona` permite chequear la sintaxis de una zona definida en un archivo de zona. Es muy útil para chequear una zona antes de incluirla en la configuración del servidor.

## 13. Servidores DNS en Windows.

La instalación del servicio DNS comienza desde el "Administración del servidor", y una vez situados en el "Panel", activando la opción *Agregar Roles y características*. Se lanzará el asistente de instalación. Pulsando *Siguiente* elegimos la opción por defecto *Instalación basada en características o roles*. A continuación elegimos sobre qué equipo servidor se instalará el servicio. Por defecto se presenta el equipo local. Al pulsar *Siguiente* aparece una lista con los roles a instalar. Marcamos la casilla *Servidor DNS*. Se presentará una nueva ventana informando de las características adicionales que se instalarán. Pulsamos en el botón *Agregar características* para cerrar la ventana y pulsamos *Siguiente* para avanzar en la instalación. Finalizamos pulsando *Siguiente* y posteriormente el botón *Instalar* para empezar la instalación del servicio.

Si desde "Herramientas administrativas" ejecutamos la consola de administración del servidor DNS, aparecerá una ventana de administración dividida en dos partes. En la parte izquierda aparecen los equipos con servicios DNS a administrar. En principio aparece nuestra máquina, pero podríamos administrar otros servidores (icono DNS, opción "Conectar con el servidor DNS" del menú contextual).

Para cada servidor se puede gestionar las zonas de resolución directa e inversa expandiendo el icono que representa el servidor. En nuestro caso no existe ninguna zona, pero las crearemos luego.

#### - Configuración del servidor DNS como solo caché.

Por defecto el servidor DNS está configurado como sólo caché (no está autorizado para ninguna zona) y responde a consultas recursivas. Podemos comprobarlo seleccionando las propiedades del servidor, y en la ventana que aparece, en la ficha "Sugerencias de raíz" se presenta una lista con los FQDN y direcciones IP de los 13 servidores DNS raíz.

Para ver las búsquedas que están en caché, se debe activar "Avanzadas" de la opción "Ver" de la barra de menú principal. Si acabamos de instalar el servicio DNS, la carpeta "Búsquedas en caché" estará vacía. Si configuramos un cliente indicándole como servidor DNS el que hemos instalado, y preguntamos por un nombre de dominio, por ejemplo, `nslookup www.google.es`, podremos comprobar como la información obtenida se queda almacenada en la caché del servidor.

#### - Configuración del servidor DNS para que reenvíe las consultas a reenviadores (*forwarding*).

En las propiedades del servidor, pestaña "Reenviadores" podemos configurar las IP de los servidores DNS a los que se reenviarán las consultas cuando se pregunte por un dominio concreto o por cualquier dominio (opción por defecto).

#### - Configuración del servidor DNS como maestro para una zona de resolución directa y una zona de resolución inversa.

Lo primero será configurar, si no lo está ya, el sufijo DNS del equipo. Se establece en propiedades de Mi PC, pestaña "Nombre de equipo", botón "Cambiar", botón "Más ...", en "sufijo principal del equipo".

(Por ejemplo *aula.izv*). Este sufijo se añadirá automáticamente al nombre del equipo cuando usemos el nombre del equipo sin indicar un dominio.

La manera más cómoda de configurar el servidor DNS como maestro para una zona de resolución directa y una zona de resolución inversa, es crear primero ambas zonas pero sin añadir registros a la zona directa hasta que no esté creada la zona inversa. Cuando añadamos los registros a la zona de resolución directa, indicaremos que se crean de forma automática los correspondientes *PTR* en la zona inversa.

A continuación vamos a definir una nueva zona de búsqueda directa gestionada por nuestro equipo, para lo cual pulsamos sobre el icono "+" mostrado junto al icono del equipo, y tras ello nos ubicamos sobre la carpeta "Zonas de búsqueda directa", pulsando en la misma con el botón derecho del ratón para elegir la opción "Zona nueva..." en el desplegable correspondiente.

Como resultado pasa a ejecutarse el asistente de creación de nueva zona, en cuya primera ventana debemos seleccionar "Zona principal". A continuación debemos indicar el nombre que vamos a asignar a la nueva zona definida y el nombre del archivo de disco donde se almacenará los datos de la zona.

Luego hemos de indicar como se realizarán las actualizaciones de nuestro servidor DNS; en nuestro caso dejaremos activada la opción "No admitir actualizaciones dinámicas". Esto obligará a actualizar manualmente los registros de recursos. Para concluir la definición de la nueva zona creada, pulsaremos sobre el botón "Finalizar".

Al pulsar en la zona, deberá aparecer una lista con los registros definidos en esta zona. En su creación se definen dos registros por defecto: *SOA* (registro de autoridad) y *NS* (servidor DNS).

Para crear la zona de resolución inversa, deberemos seguir de forma análoga los pasos anteriores hasta llegar al punto en que el asistente nos preguntará por el "Id. de red" (parte de la dirección IP que corresponde a la red, pero sin utilizar el orden inverso). Seguidamente indicaremos el nombre del archivo de disco que almacenará los datos de la zona y la opción "No admitir actualizaciones dinámicas". Cuando pulsemos el botón finalizar, se creará la zona inversa.

Al pulsar en la zona, deberá aparecer una lista con los registros definidos en esta zona. En su creación se definen dos registros por defecto: *SOA* (registro de autoridad) y *NS* (servidor DNS).

Ahora ya podemos añadir registros a las zonas. Para ello hay que seleccionar la zona y pulsar en la opción del menú principal "Acción" y elegir de la lista el tipo de registro a añadir.

Para cualquier tipo de registro creado, podemos realizar modificaciones a través de la opción "Propiedades" de su menú contextual.

Siempre que se creen o se realicen modificaciones en el servidor, deberemos reiniciar el servicio. Para ello, en el menú contextual del icono que representa al servidor, se elige la opción "Todas la tareas" y luego "Reiniciar".

## 14. Configuración de clientes.

### 14.1. Clientes Linux.

**Importante.** En Ubuntu, la configuración de las interfaces de red debe siempre realizarse a través de la misma aplicación. En Ubuntu 16 podemos usar Network Manager, o el comando *ip*. También es factible editar el archivo de configuración */etc/network/interfaces*. No se deben mezclar dichos modos por que se producen configuraciones inconsistentes.

Si optamos directamente por la edición del archivo */etc/network/interfaces* tendríamos que proporcionar los datos de todas las interfaces. Por ejemplo:

```
# the loopback network interface
auto lo
iface lo inet loopback

# the primary network interface
auto esp0s3
iface esp0s3 inet static
    address 192.168.210.211
    netmask 255.255.255.0
    gateway 192.168.210.254
    broadcast 192.168.210.255
```

Si hacemos algún cambio, siempre deberemos reiniciar el servicio de red:

```
$ sudo /etc/init.d/networking restart
$ sudo service networking restart
$ sudo systemctl restart networking.service
```

Es conveniente ejecutar posteriormente el comando *ip addr show* para verificar la configuración.

La configuración de los clientes DNS usando los archivos de configuración se realiza básicamente editando los archivos */etc/nsswitch.conf* y */etc/resolv.conf*. (**Importante. Ver anexo 1 al final del tema**).

- */etc/nsswitch.conf*. En este archivo se define el orden que usa el *resolver* a la hora de buscar información sobre nombres de dominio. El orden definido es: primero *files* (se consulta en el archivo */etc/hosts*) y luego *dns* (se consulta a los servidores DNS definidos en */etc/resolv.conf*).
- */etc/resolv.conf*. Este archivo contiene una serie de atributos a los que se puede asociar valores. En las distribuciones de Ubuntu más recientes es en realidad un link a */run/resolvconf/resolv.conf*.
  - *nameserver*. Indica los servidores DNS que consultará el resolver. Se pueden indicar varios.
  - *domain*. Especifica el dominio local por defecto al que pertenece el equipo y el que se añadirá cuando se realice una búsqueda cuando el nombre de dominio no esté cualificado.
  - *search*. Si un equipo pertenece a varios dominios permite ampliar la lista de dominios que se añadirán a los nombres de dominio en las búsquedas de nombres no cualificados.

Las opciones *domain* y *search* no pueden usarse simultáneamente. Por ejemplo:

```
nameserver 192.168.210.211
nameserver 8.8.8.8
search aula.izv
```

Si usamos la aplicación *Network Manager*, desde ajustes de IPv4 de las propiedades TCP/IP de la conexión de red podremos editar en los correspondientes cuadros de texto tanto las IP de los servidores DNS (se escriben uno detrás de otro separados por comas), como el dominio de búsqueda (*search*) al que pertenece la máquina. La aplicación *Network Manager* guarda los datos de configuración en */etc/NetworkManager/system-connections/nombre-de-la-conexión*.

Es conveniente editar el archivo */etc/hostname* y comprobar que contiene el nombre que realmente queremos para el equipo.

## 14.2. Clientes Windows.

En los clientes Windows la configuración de red se establece a través de las propiedades del protocolo TCP/IP que podemos activar mediante las propiedades de la “Conexión de área local”. En “Usar las siguientes direcciones de servidores DNS” es posible definir dos servidores DNS. El segundo servidor DNS que señalemos, será un servidor alternativo en el caso de que el primero de ellos no esté disponible.

También en “Opciones avanzadas”, en la pestaña “DNS” se pueden añadir más servidores DNS alternativos, y sufijos para los nombres DNS no cualificados. Por defecto el único prefijo que se añade a los nombres no cualificados es el llamado sufijo principal que se configura en las propiedades del nombre del equipo (Equipo o MiPc -> Propiedades -> Configuración avanzada del sistema -> Pestaña Nombre de equipo -> botón Cambiar -> Botón Más -> sufijo DNS principal del equipo). Si se desea añadir más sufijos al nombre del equipo, habrá que habilitar el botón de opción “Anexar estos sufijos (en este orden)” e introducir la lista de sufijos.

El comando **ipconfig /displaydns** permite ver el contenido de la caché de resolución de un cliente DNS, incluyendo las entradas cargadas previamente desde el archivo *hosts* local, así como los registros de recursos obtenidos recientemente para las consultas de nombres que han sido resueltas.

El comando **ipconfig /flushdns** permite vaciar el contenido de la caché de resolución de un cliente DNS. Al restablecer la caché no se eliminan las entradas que se cargan previamente del archivo *hosts* local. Para eliminar estas entradas de la caché, habría que eliminarlas del archivo *hosts*.

## 15. Herramientas de consulta.

Los comandos *nslookup*, *dig* y *host* permiten realizar consultas a servidores DNS. Estos comandos permiten comprobar el funcionamiento del servicio y de los servidores DNS. El comando *nslookup* es más antiguo y ofrece menos información que *dig*, pero está disponible en equipos Linux y Windows.

- **nslookup.** De forma general obtiene registros de recursos tipo A o PTR, pero dispone de opciones para obtener otros tipos de informaciones. Puede funcionar en línea de comandos o de forma interactiva. El comando actúa de una forma predeterminada según se haya establecido mediante unos parámetros, los cuales podemos cambiarlos. Esta lista de parámetros se pueden obtener invocando el comando seguido del argumento *-all*.

El formato para usar nslookup mediante línea de comandos es:

```
nslookup [-type=tipo registro] [nombre dominio] [servidorDNS]
```

Ejemplos de uso en línea de comandos:

```
# muestra cual es el servidor DNS configurado en la interfaz de red. (Servidor predeterminado para nslookup)
nslookup
```

```
# obtiene la dirección (por defecto -type=A) asociada al nombre de dominio www.mec.es (pregunta al servidor
#predeterminado
nslookup www.mec.es
```

```
# obtiene la dirección (por defecto -type=A) asociada a un nombre de dominio preguntando al servidor a.nic.es
nslookup www.mec.es a.nic.es
```

```
# obtiene el nombre de dominio (por defecto -type=PTR) correspondiente a la IP
nslookup 150.214.20.1
```

```
# obtiene todos los tipos de registros del dominio mec.es
nslookup -type=ANY mec.es
```

```
# obtiene todos los servidores de nombres (NS) para el dominio mec.es preguntando al servidor 8.8.8.8
nslookup -type=NS mec.es 8.8.8.8
```

```
# obtiene el registro SOA del dominio example.com (pregunta al servidor predeterminado)
nslookup -type=SOA example.com
```

```
# obtiene el registro SRV del servicio ldap del dominio example.com (pregunta al servidor predeterminado)
nslookup -type=SRV _ldap_tcp.example.com
```

Cuando se usa de forma interactiva (sin argumentos) aparece un indicador de órdenes (>) donde se puede ir escribiendo comandos. Se sale con “exit” o bien Ctrl+C (Windows) o Ctrl+D (Linux). Las posibles órdenes que se pueden ejecutar dependen del sistema operativo, algunas son comunes a Windows y a Linux, y otras sólo funcionan en uno de ellos.

Con unos ejemplos veremos las posibilidades:

```
# configura el servidor a quien se le preguntará y se le pregunta por los servidores autorizados para ugr.es
nslookup
> server 8.8.8.8
> set type=NS
> ugr.es
```

```
# configura el servidor a quien se le preguntará y se pregunta por el nombre de dominio correspondiente a la IP
nslookup
> server a.nic.es
> set type=PTR
> 150.214.20.1
```

- **dig.** La herramienta *dig* ofrece más opciones y suministra más información en sus respuestas que *nslookup*. Organiza la información en secciones que se corresponden con los campos principales de los mensajes DNS. Los registros de recursos los muestra con un formato similar a como se escriben en los archivos de zona.

Tradicionalmente *dig* sólo ha estado disponible en sistemas Unix/Linux, pero actualmente existen versiones para Windows; incluso como aplicación con interfaz gráfica, como es el caso de *Ezdig*, aplicación *freeware* de la empresa *Eztk* (<http://www.eztk.com/>).



El formato (simplificado) es el siguiente:

```
dig [@dns] domain [question-type]
```

donde *dns* especifica el servidor de dominio al que se le realizará la consulta; *domain* indica el nombre de dominio a preguntar, y *question-type* expresa el tipo de recurso consultado. Ejemplo:

```
# pregunta al servidor predeterminado por la IP correspondiente al nombre del dominio www.mec.es (Por defecto
# question-type toma el valor A)
dig www.mec.es
```

```
# pregunta al servidor predeterminado por el nombre de dominio asociado a la IP 8.8.4.4
dig -x 8.8.4.4
```

```
# pregunta al servidor 8.8.8.8 por el nombre del dominio mostrando la traza de la consulta
dig @8.8.8.8 www.google.es +trace
```

```
#pregunta al servidor 8.8.8.8 por los servidores autorizados para el dominio es
dig @8.8.8.8 es NS
```

```
#pregunta al servidor 1.1.1.1 por el nombre canónico del dominio ftp.uniovi.es
dig @1.1.1.1 ftp.uniovi.es CNAME
```

```
#pregunta al servidor predeterminado por los servidores de correo del dominio uniovi.es
dig uniovi.es MX
```

```
# pregunta al servidor 8.8.8.8 por todos los registros de recursos del dominio mec.es
dig @8.8.8.8 mec.es ANY
```

```
# obtiene el registro SRV del servicio ldap del dominio example.com (pregunta al servidor predeterminado)
dig _ldap_tcp.example.com SRV
```

- **host.** Es un comando Linux cuyo formato simplificado es:

```
host [options] name [server]
```

donde *options* indica entre otras cosas el tipo de recurso a preguntar, *name* es el nombre de dominio por el que se consulta y *server* es el servidor de dominio consultado. Ejemplos:

```
#obtiene la IP correspondiente al nombre del dominio (registro tipo A)
host www.mec.es
```

```
#obtiene la dirección IP correspondiente al nombre de dominio (registro PTR)
host 8.8.4.4
```

```
#pregunta a 8.8.8.8 por los servidores autorizados para el dominio (registros NS)
host -t NS es 8.8.8.8
```

```
# pregunta a 8.8.8.8 por todos los registros de recursos del dominio con información detallada en secciones
host -v -t ANY mec.es 8.8.8.8
```

## 16. Anexo 1. El resolutor en Ubuntu 17 y posteriores.

Tradicionalmente, para realizar la resolución de nombres en los sistemas UNIX, se usa el servicio Name Service Switch (NSS). Éste servicio forma parte de la biblioteca *glibc* escrita en C de los sistemas operativos UNIX basados en GNU. Esta biblioteca contiene la API (código) que se utiliza para resolver nombres de dominio.

Con NSS se permite que ciertos datos del sistema sean proporcionadas por servicios diferentes, cuyo orden de preferencia puede ser configurado por el administrador en el archivo `/etc/nsswitch.conf`. Así por ejemplo, los datos responsable de la resolución de nombres de dominio que se indican en `/etc/nsswitch.conf` podrían ser:

```
hosts:    files mdns4_minimal [NOTFOUND = return] dns myhostname
```

En este caso, la resolución empezaría buscado en el archivo `/etc/hosts`, a continuación usaría el servicio mDNS (multicast Domain Name System), proseguiría con el servicio DNS, y por último se emplearía el método implementado por el *plugin* para NSS denominado *myhostname*.

*El servicio mDNS permite resolver nombres de dominio en redes pequeñas que no incluyen un servicio DNS. Cuando un cliente mDNS necesita resolver un nombre de dominio, envía un mensaje de multidifusión en el que pide al equipo que tenga ese nombre se identifique. La máquina que tiene dicho nombre envía un mensaje también de multidifusión con su IP.*

*De manera predeterminada, mDNS resuelve única y exclusivamente nombres de host que terminan en el dominio de nivel superior (TLD) .local. Existen diferentes implementaciones, siendo las más conocidas Bonjour (Apple) y Avahi (GNU/Linux y BSD).*

*El sistema de resolución implementado como plugin para NSS denominado **myhostname** intenta resolver el nombre de dominio examinando el contenido del archivo /etc/hostname.*

Pero actualmente, las nuevas distribuciones de Linux que usan *systemd* (sistema de inicio de los sistemas Linux que sustituye al sistema de inicio *System V init* tradicional), proporcionan un servicio denominado *systemd-resolved* que ofrece el servicio DNS a algunas aplicaciones locales (como por ejemplo a ciertos navegadores). Este servicio tiene 4 formas de manejar el archivo /etc/resolv.conf que es el que usa el *resolutor* proporcionado por la librería *glibc*. En particular dos de ellas son:

1. El modo en el que *systemd-resolved* es en realidad un cliente de /etc/resolv.conf. Es decir, en este modo se conserva la información que contenga el archivo /etc/resolv.conf de forma que los servidores de nombres y dominio de búsqueda contenidos en dicho archivo son los que usará el servicio DNS proporcionado por la librería *glibc* y que anteriormente hemos comentado.

Este sistema sería el adecuado cuando el archivo /etc/resolv.conf es modificado directamente por el administrador o bien por otros programas, como es el caso de la aplicación NetworkManager, o cuando usamos una configuración proporcionada por el servicio DHCP para las interfaces de red.

2. El segundo modo es el modo de operación recomendado por *systemd-resolved* e **implementado por defecto pero no por ello siempre el más conveniente**. Consiste en usar un único servidor DNS implementado de forma local en la interfaz de red 127.0.0.53, (*nameserver 127.0.0.53*) al cual normalmente se le acompaña con una lista de dominios de búsqueda (*search dominio*).

La información de dicho servidor DNS la podemos ver si seguimos el *link* desde /etc/resolv.conf al archivo enlazado /run/systemd/resolve/stub\_resolve.conf.

En este modo, los clientes locales que leen el archivo /etc/resolv.conf realmente se dirigen a este servidor DNS local en lugar de usar el sistema DNS tradicional.

En realidad este DNS es muy adecuado cuando queremos resolver nombres de dominio locales, pero siempre que dichos dominios terminen su nombre de dominio de nivel superior (TLD) con .local, .intranet, .net, .private (ver salida del comando `systemctl status systemd-resolved`). Si usamos otros nombres de dominio locales que no sean éstos, como suele suceder en muchas ocasiones, este servidor DNS intentará buscarlos redirigiendo las preguntas hacia el exterior, y evidentemente no los encontrará, a no ser que añadamos dichos dominios de búsqueda mediante la opción *search dominio*.

Por lo tanto, si queremos resolver nombres de dominio locales que no usen los nombres de dominio anteriores (.local, .intranet, .net, .private) tendríamos que modificar manualmente el archivo /etc/resolv.conf con la información que desea el administrador o bien dejar que sean las propias aplicaciones que lo necesiten las que modifiquen dicho archivo.

El problema de esta solución es que en el siguiente reinicio, el sistema volverá a la configuración por defecto (modo 2) donde /etc/resolv.conf vuelve a ser enlazado con el archivo /run/systemd/resolve/stub\_resolve.conf.

Para resolver este problema, debemos hacer que *systemd-resolved* opere en el tradicional modo 1, en vez del modo 2. Para ello el servicio *systemd-resolved* debe usar el contenido del archivo /run/systemd/resolve/resolv.conf en lugar del contenido en /run/systemd/resolve/stub\_resolve.conf

Para ello eliminamos el enlace de /etc/resolv.conf a /run/systemd/resolve/stub\_resolve.conf y creamos uno nuevo al archivo /run/systemd/resolve/resolv.conf. Es decir:

```
usuario@localhost:~# sudo ls -la /etc/resolv.conf
lrwxrwxrwx 1 root root 29 mar 7 20:20 /etc/resolv.conf -> ../run/systemd/resolve/stub_resolve.conf

usuario@localhost:~# sudo rm -f /etc/resolv.conf
usuario@localhost:~# sudo ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf

usuario@localhost:~# sudo ls -la /etc/resolv.conf
lrwxrwxrwx 1 root root 32 mar 8 07:30 /etc/resolv.conf -> /run/systemd/resolve/resolv.conf
```