Unidad 2

Servicio de resolución de nombres de dominio

Servicios de red e internet

Índice

Introducción

DNS son las siglas de Domain Name Service (Servidor de nombres de dominio), y es el protocolo encargado de asociar un nombre en concreto con una dirección IP con la intención de poder localizar un host sin necesidad de escribir su dirección IP completa.

Las funciones básicas que realiza el DNS son:

- Conversión de nombres de dominio a direcciones IP.
- Localización de los servidores de correo dentro de cada dominio.

La primera cuestión por la que es tan usado DNS es porque un nombre es mucho más fácil de interpretar y recordar para los usuarios que las direcciones IP, y, además, es posible que esta IP cambie, mientras que el nombre de dominio es improbable, lo que lo hace aún más fiable.

Al finalizar esta unidad

- Seremos capaces de identificar y describir escenarios en los que surge la necesidad de un servicio de resolución de nombres.
- Podremos clasificar los principales mecanismos de resolución de nombres.
- Sabremos la descripción de la estructura, nomenclatura y funcionalidad de los sistemas de nombres jerárquicos.
- Conoceremos como instalar y configurar servicios jerárquicos de resolución de nombres.
- Podremos preparar el servicio para el reenvío de consultas de recursos externos a otro servidor de nombres.
- Sabremos organizar el servicio para almacenar y distribuir las respuestas procedentes de otros servidores.
- Conoceremos como añadir registros de nombres correspondientes a una zona nueva, con opciones relativas a servidores de correo y alias.
- Seremos capaces de implementar soluciones de servidores de nombres en direcciones IP dinámicas.
- Sabremos como realizar transferencias de zona entre dos o más servidores.

2.1. Sistemas de nombres planos y jerárquicos

Cuando dos equipos inician la comunicación de la red, siempre se tienen que indicar tanto el origen como el destino. En este caso, los paquetes deben de llevar la dirección MAC de origen y destino, la dirección IP de origen y destino y los puertos de origen y destino. Como todo tiene que ser siempre así, eso quiere decir que si nosotros buscamos en Internet *universae.com*, realmente estamos en busca de conectarnos con una dirección IP específica. Esto se consigue a través del servicio DNS.

Este servicio, que es el servicio de nombres de dominio, o DNS (*domain name service*) es el que se encarga de asignar las direcciones IP con un nombre en lenguaje escrito. Al igual que hace dicha asociación, también es el encargado de averiguar cuál es este par en una red externa, de modo que, si solicitamos un recurso y especificamos el nombre, nos devolverá la IP, o al revés.

La principal ventaja del servicio DNS es que hace las direcciones mucho más legibles, con lo que ganamos en sencillez y comodidad, porque por dirección IP también se podría acceder a una página web.

Un ejemplo de como funcionan los servidores de DNS lo podríamos realizar lanzando el comando `ping`.

Si lanzamos este comando por ejemplo a *Ubuntu.com*, veremos que aparece la dirección IP más abajo.

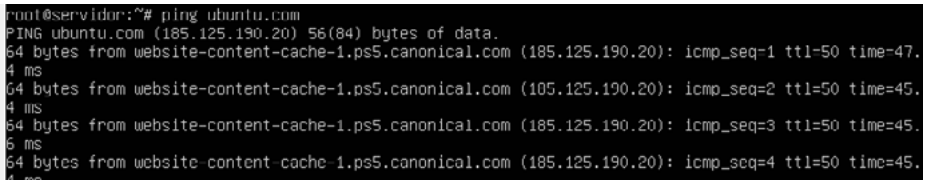

Imagen 1. ping

Y si buscamos esa dirección IP en un navegador web, vemos que efectivamente entramos a dicha página.

Tenemos dos modos de uso de DNS reales:

- **DNS locales:** son los servidores DNS que se alojan en nuestra red local. Este servicio se queda dentro de nuestra red local, por lo que el tráfico hacia el exterior se elimina y evitamos que el almacén de consultas genere cuellos de botella.
- **DNS remotos:** son los alojados en internet y que se usan cuando no tenemos DNS locales. Por ejemplo, es muy común el uso de DNS de Google (8.8.8.8 y 8.8.4.4).

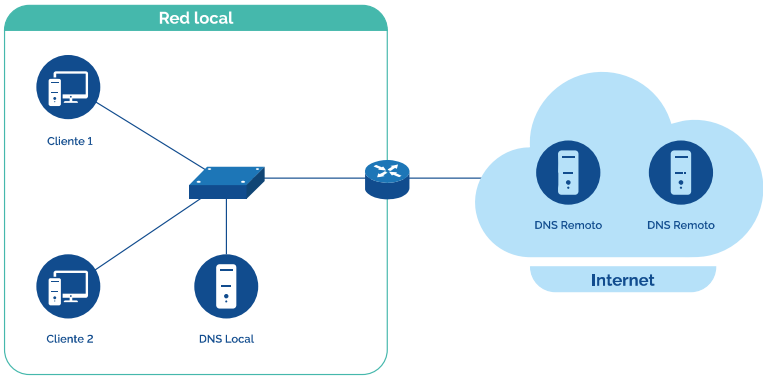

Imagen 2. DNS local y remoto

En una red, tenemos dos modos de nombrar a los equipos, independientemente de si son locales o remotos:

- **Nombres planos.** El nombre sirve para la identificación de un elemento dentro de un conjunto, sin más información que sirva de descripción adicional.
- **Nombres jerárquicos.** El nombre se compone por valores que van indicando información adicional. Los nombres jerárquicos en DNS usan el punto "." Para esta separación, y es lo común en las páginas Web, por ejemplo *drive.google.es.*

En el servicio de resolución de nombres, lo más normal es usar nombres jerárquicos, donde el nombre completo de un equipo indicará la rama a la que pertenece y su marco legislativo.

2.2. Servidores raíz y dominios de primer nivel y sucesivos

Cuando establecemos una jerarquía de nombres, esta siempre se basa en una estructura en árbol, al igual que pasa en los sistemas de directorios. Si queremos nombrar un equipo, iremos de más a menos por la estructura pasando por tres niveles principales, el dominio raíz, el de primer nivel y los niveles inferiores. Cada nivel lo administra una entidad reguladora que estable reglas para el uso de los nombres.

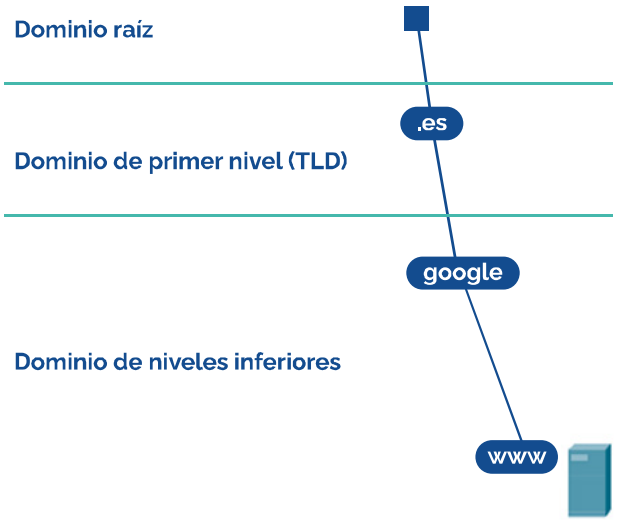

Imagen 3. Estructura jerárquica de nombres de dominio

Vamos ahora a detallar estos niveles:

- **Dominio raíz:** aunque en la figura anterior lo hemos representado como un cuadrado, es un punto "." y sirve de partida para todos los nombres del dominio. La autoridad encargada de sus nombres y de las direcciones IP asociadas en Internet es la IANA, cuya función es la coordinación de los DNS de todo el mundo, además de asegurarse de su funcionamiento.
- **Dominios de nivel superior (TLD):** son los que corresponden a organizaciones, empresas o localizaciones geográficas, dos ejemplos pueden ser, *COM*, para entidades comerciales o el *código del país*, como puede ser *es* para España.
- **Dominios de niveles inferiores:** son las categorías y elementos concretos que especifican una organización u organismo concreto dentro del dominio de primer nivel.

Las *zonas* o espacios de nombres de domino son los nombres contiguos que se asocian a una parte del árbol jerárquico.

El conjunto de nombres específico es llamado *registro de recursos o RR*, y se aloja en uno o varios servidores DNS conteniendo las asociaciones nombre-dirección IP. Un servidor está autorizado sobre una zona cuando contiene un RR de una zona específica.

Se pueden descentralizar los RR por parte de un servidor maestro delegando responsabilidades sobre las zonas a otros servidores llamados esclavos o *slaves*.

Por lo general, en los niveles más bajos del árbol encontramos el nombre del equipo o el servicio que se ofrece, como puede ser FTP para la transferencia de archivos, o *debian1* como nombre de equipo.

En el ejemplo de nuestra imagen, el nombre completo sería empezando de arriba abajo, y quedaría *www.google.es.*, lo que se conoce como *nombre FQDN (Fully Qualified Domain Name)*. Aunque el nombre siempre termina en punto, desde hace ya un tiempo, esto puede ser omitido y aun así funciona.

2.3. Resolutores de nombres: proceso de resolución de un nombre de dominio

El *resolutor o resolver*, es la rutina del sistema que hace de intermediario entre aplicaciones, sistema y servidores DNS. Si o el sistema o una aplicación en concreto necesita información acerca de nombre o dirección IP, llama al resolutor pidiendo de forma entendible la información deseada.

2.3.1. Protocolo DNS y proceso de resolución

Lo primero que necesitamos aclarar para comprender de manera correcta como funciona la resolución de nombres, es el concepto de mensaje de consulta, además de los campos que lo componen.

Los mensajes de consulta son paquetes enviados por parte del cliente al servidor con la intención de obtener, o bien una dirección IP, o bien un nombre de dominio. El protocolo que se usa es DNS (*Domain Name System)* que se establece en la capa de aplicación y se encuentra regulado por la RFC 1034 y la RF 1035. Los paquetes son UDP, sin conexión y el puerto de escucha es el 53. El formato de las consultas y respuestas es predefinido y su tamaño varía dependiendo de e la respuesta.

Los campos de los mensajes son los del siguiente cuadro:

Campos de mensajes DNS	
Cabecera	Es el campo que contiene los *flags indicadores*. No se deben de mezclar los *flags* con nombres de campos siguientes, puesto que estos detallan más información del proceso.
Pregunta	Contiene el nombre de la pregunta y parámetros adicionales.
Respuesta	Contiene los RR que sirven de respuesta a la pregunta inicial.
Autoridad	Indica si los RR tienen o no autoridad en la resolución de mensajes.
Adicional	Contiene todos los datos adicionales de la consulta.

2.3.2. Tipos de consultas

Si tenemos en cuenta quien resuelve la petición, tenemos dos tipos de consultas:

- **Consulta recursiva.** El resolutor lanza una consulta de un nombre completo y esta suele ser desde un cliente hacia el servidor. En este tipo de consultas, el cliente lanza al servidor DNS una consulta de nombre FQDN con la intención de recibir la dirección IP. El servidor, si no la tiene almacenada en caché o en su RR, la buscará en otros servidores DNS con la intención de aportar respuesta, aunque sea negativa, es decir, que no existe o no se ha podido obtener.
- **Consulta iterativa.** Suele llevarse a cabo de servidor a servidor y la respuesta la solemos recibir de manera parcial. Habrá que juntar todas las respuestas recibidas de los distintos servidores y se obtendrá la que entregaremos al cliente.

La siguiente ilustración, nos muestra gráficamente el recorrido que se haría en caso de solicitar en un navegador Web el acceso a *www.google.es*.

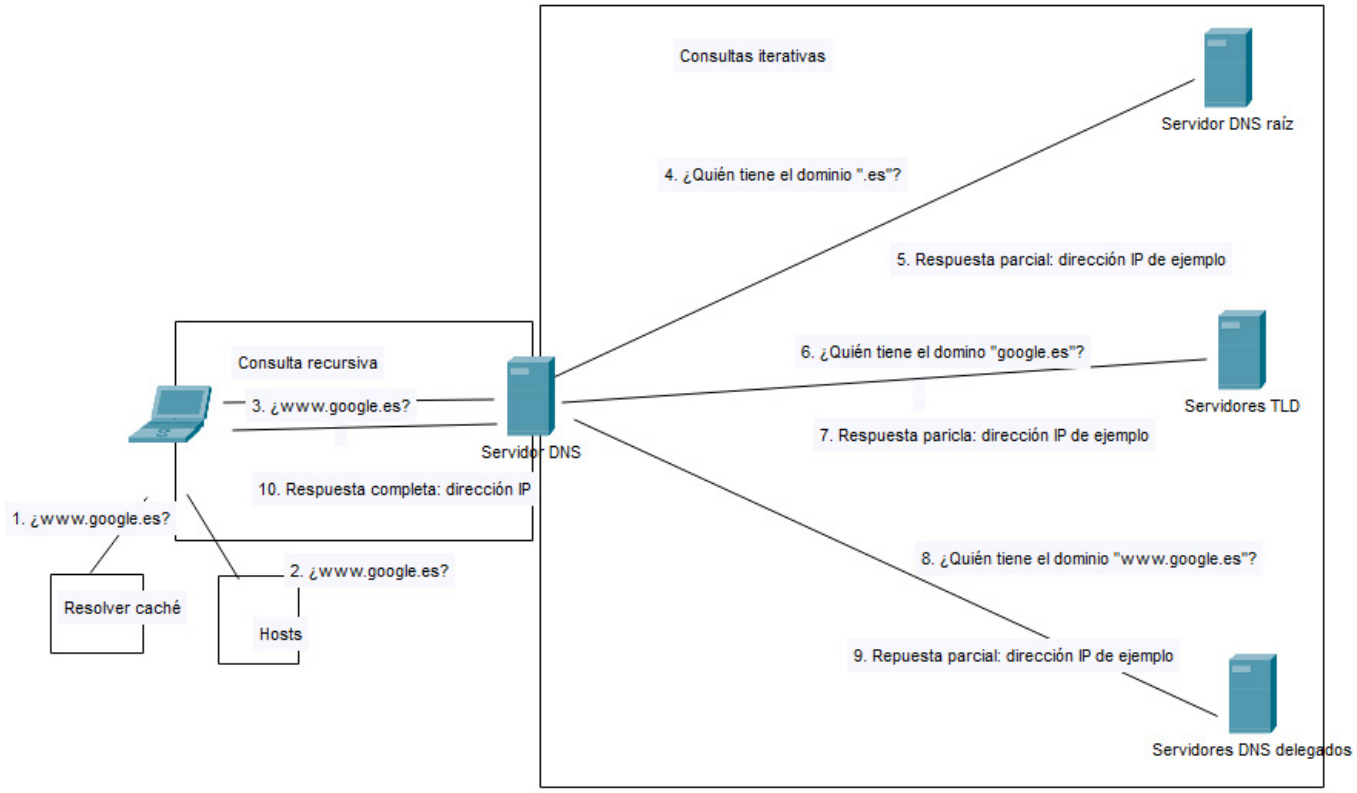

Imagen 4. Proceso de resolución de consultas de DNS recursivas e iterativas

Antes de consultar a los DNS, los clientes comprueban si las direcciones IP asociadas a nombres se encuentran almacenadas en la caché o en el fichero *hosts*.

El fichero *hosts* se encuentra en distintos sitios dependiendo de si estamos en un sistema u otro:

- En Windows: *C:\Windows\System32\drivers\etc\hosts*.

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

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
```

Imagen 3. Fichero hosts en Windows

- En Linux: */etc/hosts*:

```
profesor@servidor:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 servidor

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

Imagen 4. Fichero hosts en Linux

Un ejemplo de las consultas que hacen los distintos servidores lo podemos obtener con el comando `dig`. Este comando se encuentra usable tanto en Windows como en Linux y si lo ejecutamos del siguiente modo,

```
dig +trace www.google.es
```

Podemos ver que nos da la siguiente salida:

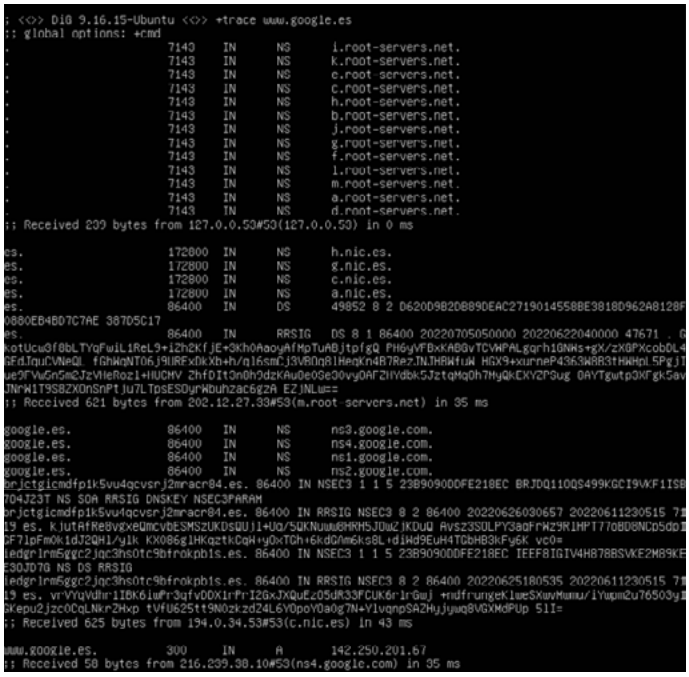

Imagen 5. Comando dig

Podemos ver que nuestro servidor DNS interno y directo es el 127.0.0.53, y que se van haciendo numerosas consultas a distintos servidores para comprobar los dominios parcialmente, al igual que en el ejemplo anterior, obteniendo finalmente que la dirección de *www.google.es* es 142.250.201.67.

En la imagen anterior, también se observa, que *google.es* nos lo devuelven los propios servidores de Google.

2.3.3. Tipos de resoluciones: directa e inversa

Tenemos dos tipos de resoluciones:

- **Resolución directa:** esta se da cuando un cliente lanza una consulta sobre un nombre y lo que quiere es obtener una dirección IP. Es la resolución por defecto y la más usada.
- **Resolución inversa:** el cliente solicita un nombre al aportar una dirección IP. Esto es usado sobre todo por aplicaciones que, por seguridad, lo hacen de este modo con la intención de detectar *malwares* Se denominan *PTR* y se definen en el fichero de zona como *RR reverse.*

Todo Servidor DNS debe de funcionar con la resolución directa, pero, aunque la inversa es opcional, debemos de saberla porque es recomendable.

2.4. Instalación de BIND

El *software* que vamos a usar para la gestión del servicio va a ser BIND, que es el más usado y se encuentra gestionado por el ISC. La versión actual que opera sobre *Ubuntu Server* es la 9. Lo ideal para la configuración del servicio es que la dirección de red del servidor sea estática, porque con una dirección dinámica, va a fallar si se le asigna al servidor una diferente cada vez.

Entonces, para la instalación de este *software*, debemos de hacer lo siguiente:

1. Entramos en el terminal y nos convertimos en administradores, es decir, *root*.
2. Lo primero que debemos de hacer es actualizar los repositorios de Ubuntu y los paquetes.
3. Una vez que lo hemos hecho, procedemos con la instalación del *software*. El comando para usar es:

   ```
   apt install bind9
   ```

```
root@servidor:~# apt install bind9
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libfwupdplugin1
Utilice «apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  bind9-utils dns-root-data python3-ply
Paquetes sugeridos:
  bind-doc resolvconf python-ply-doc
Se instalarán los siguientes paquetes NUEVOS:
  bind9 bind9-utils dns-root-data python3-ply
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 489 kB de archivos.
Se utilizarán 1.943 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s_
```

Imagen 6. Instalación de *bind9*

4. Aunque no es obligatorio, para un rendimiento optimo podemos instalar además el conjunto de herramientas DNS adicionales, llamadas las *dnsutils*.

```
root@servidor:~# apt install dnsutils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
```

Imagen 7. Instalación de *dnsutils*

5. Cuando tenemos ambos paquetes instalados, lanzamos `sudo reboot` para reiniciar el equipo.
6. Una vez se ha reinicio el equipo, comprobamos que el servicio se encuentra activo con el comando:

   ```
   systemctl status bind9
   ```

```
root@servidor:~# systemctl status bind9
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-06-22 08:38:45 UTC; 2min 16s ago
       Docs: man:named(8)
    Process: 673 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 733 (named)
      Tasks: 5 (limit: 2289)
     Memory: 25.4M
        CPU: 48ms
     CGroup: /system.slice/named.service
             └─733 /usr/sbin/named -u bind

jun 22 08:38:45 servidor named[733]: network unreachable resolving './DNSKEY/IN': 2001:7fe::53#53
jun 22 08:38:45 servidor named[733]: network unreachable resolving './NS/IN': 2001:7fe::53#53
jun 22 08:38:45 servidor named[733]: network unreachable resolving './DNSKEY/IN': 2001:500:2d::d#53
jun 22 08:38:45 servidor named[733]: network unreachable resolving './NS/IN': 2001:500:2d::d#53
jun 22 08:38:45 servidor named[733]: network unreachable resolving './DNSKEY/IN': 2001:500:2f::f#53
jun 22 08:38:45 servidor named[733]: network unreachable resolving './NS/IN': 2001:500:2f::f#53
jun 22 08:38:45 servidor named[733]: network unreachable resolving './DNSKEY/IN': 2001:503:c27::2:3>
jun 22 08:38:45 servidor named[733]: network unreachable resolving './NS/IN': 2001:503:c27::2:30#53
jun 22 08:38:45 servidor named[733]: managed-keys-zone: Key 20326 for zone . is now trusted (accept>
jun 22 08:38:45 servidor named[733]: resolver priming query complete
lines 1-22/22 (END)_
```

Imagen 8. Estado de *bind9*

Una vez que se han instalado ya todos los paquetes, tenemos los ficheros de configuración del servicio ubicados en la ruta */etc/bind*.

```
root@servidor:~# cd /etc/bind
root@servidor:/etc/bind# ls
bind.keys  db.127  db.empty  named.conf                named.conf.local    rndc.key
db.0       db.255  db.local  named.conf.default-zones  named.conf.options  zones.rfc1918
root@servidor:/etc/bind# _
```

Imagen 9. */etc/bind*

De esta ruta, podemos destacar dos ficheros:

- **Named.conf:** el principal fichero para la configuración.
- **Named.conf.options:** incluimos aquí el valor del directorio donde vamos a localizar los ficheros de configuración y almacenamos el valor para el reenvío de las consultas que no se puedan resolver.

Los demás ficheros se explicarán conforme vayamos avanzando.

2.5. Tipos de registros

Llamamos zonas a los espacios de nombres que almacenamos en la base de datos del servidor formada por varios *registros de recursos*.

Los ficheros que definen las zonas tienen, además, cadenas de texto que se usan para la definición de ciertos valores dentro de una zona. Cada cadena puede ser de un tipo dsitinto, que son:

- **Comentarios.** Comienzan con el punto y coma *";"* y se usan para incluir aclaraciones sobre el fichero.
- **Directivas.** Se usan para especificar aspectos del RR. Siempre empiezan por el símbolo del dólar *"$"* y se usan, sobre todo:
  - **$ORIGIN.** Nos define el nombre del dominio que se va a incluir al final de cualquiera de los nombres que definamos en los RR y que, además, no sea el acabado en punto ".". La directiva no es obligatoria, pues como veremos más adelante, se puede usar una definida previamente.
  - **$TTL.** Es el valor del *Time to Live* de una zona en concreto. Se expresa en segundos y como cada recurso puede tener el suyo propio, tampoco tenemos por qué especificarlo.
- **Registro de recurso.** Es la cadena de texto que se usa para definir las entidades dentro de nuestro dominio. Lo que más se suele usar es:
  - **SOA.** Es el registro que define donde comienza una zona. Se debe colocar inmediatamente después de las directivas y nos ayuda a definir información muy importante acerca de la autoridad de los RR para cada zona. Su sintaxis es:

```
@ IN SOA servidor_DNS_primario email_administración (

        Número_serie

        Tiempo_refresco

        Tiempo_reintento

        Tiempo_de expiración

        TTL_mínimo )
```

  En el siguiente ejemplo podemos ver la zona de localhost definida por defecto.

```
@       IN      SOA     localhost. root.localhost. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
```

Imagen 10. Registro *SOA*

» **NS.** Registro suado para definir los nombres de los servidores autorizados para gestionar una zona concreta. Su sintaxis es:

```
IN NS nombre_servidor
```

Tenemos aquí el servidor *localhost* como ejemplo:

```
@       IN      NS      localhost.
```

Imagen 11. Registro *NS*

» **A.** Se usa para asignación de una dirección IPv4 con un nombre. La sintaxis es:

```
Host IN A IP
```

En el siguiente ejemplo se ve la relación (sin *host*) de la IP de *localhost*:

```
@       IN      A       127.0.0.1
```

Imagen 12. Registro A

» **AAAA.** Define este registro la asignación de una dirección Ipv6 con un nombre. Su sintaxis es:

```
Host IN AAAA IP
```

De nuevo el ejemplo con *localhost*:

```
@       IN      AAAA    ::1
```

Imagen 13. Registro AAAA

» **MX.** Es el registro usado para la definición de los servidores de correo electrónico para el dominio. Su sintaxis:

```
IN MX valor_preferencia nombre_servidor_correo
```

En este caso hay que mencionar que *valor_preferencia* hace referencia al orden de preferencia si tenemos varios servidores de correo electrónico para una misma zona.

» **CNAME.** Es el nombre canónico que se usa para definir alias a los nombres de dominio, es decir, se usa para que una misma dirección pueda tener varios nombres. Para eso debe estar declarado primeramente el nombre, claro está. Su sintaxis es:

```
Alias IN CNAME nombre_dominio
```

2.6. Zonas primarias y secundarias

Ya hemos comentado lo que era una zona. Bien, sabiendo esto, existe la posibilidad de que haya más de un servidor DNS, y que uno actúe como maestro o primario y lñosde más sean secundarios o esclavos. Pues hay que saber que la *zona primaria* es la que se define en el servidor DNS primario y las *zonas secundarias* son las copias de la zona primara que se realizan en los DNS secundarios.

Si un servidor DNS gestiona las asociaciones de nombres con direcciones de una zona en concreto, este está a*utorizado* sobre esa zona y por tanto todas las respuestas que proporcione dicho servidor sobre nombres definidos en su zona, serán respuestas *autoritativas*.

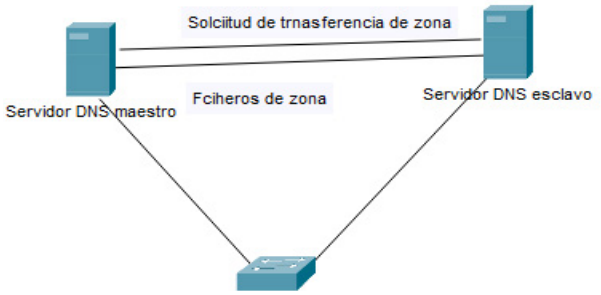

Imagen 14. Transferencia de zona

El procedimiento se debe de configurar para que sea automático.

Tenemos tres métodos principales de configuración del servicio, que son:

- **DNS caché y fordwarder:** en esta manera el servidor buscará las resoluciones en la memoria caché y en caso de no encontrarlas, las solicitará a los demás servidores para iniciar el proceso iterativo de resolución de consultas.
- **DNS primario maestro:** en esta manera el servidor tiene una definición de la zona, que almacena en el archivo de RR. Este servidor es el autorizado para dicha zona.
- **DNS secundario maestro:** el servidor recibe las transferencias de las zonas de otros servidores DNS primarios.

La clasificación no es excluyente, es decir, los servidores pueden ser cualquiera de los tres e incluso los tres simultáneamente.

2.6.1. Preparación del servicio para almacenamiento y reenvío (caché y fordwarder)

Este método de resolución de nombres es la configuración que, por defecto, tiene *BIND* una vez instalado. Nuestro servidor almacenará las respuestas en caché y si no puede dar respuesta a alguna petición, redirecciona estas hacia servidores externos.

La caché se almacena siempre 7 días por defecto. Si queremos ver el contenido de la caché, debemos de volcarla en un fichero de texto con el siguiente comando, siempre como *root*:

```
rndc dumpdb -cache
```

```
root@servidor:~# rndc dumpdb -cache
root@servidor:~#
```

Imagen 15. Traducir la caché de DNS a texto plano

Si el comando se ha ejecutado sin error alguno, el fichero se ha guardado en la ruta */var/cache/bind* y su nombre es *named_dump.db* y lo podríamos visualizar con cualquier comando como puede ser `cat`.

```
;
; Start view _default
;
;
; Cache dump of view '_default' (cache _default)
;
; using a 86400 second stale ttl
$DATE 20220622065637
; secure
.                       599076  IN NS   a.root-servers.net.
                        599076  IN NS   b.root-servers.net.
                        599076  IN NS   c.root-servers.net.
                        599076  IN NS   d.root-servers.net.
                        599076  IN NS   e.root-servers.net.
                        599076  IN NS   f.root-servers.net.
                        599076  IN NS   g.root-servers.net.
                        599076  IN NS   h.root-servers.net.
                        599076  IN NS   i.root-servers.net.
                        599076  IN NS   j.root-servers.net.
                        599076  IN NS   k.root-servers.net.
                        599076  IN NS   l.root-servers.net.
                        599076  IN NS   m.root-servers.net.
; secure
                        599076  RRSIG   NS 8 0 518400 (
                                        20220706050000 20220623040000 47671 .
                                        mOVSeCNfR3cgYgqb162gUNP8p+ZaEW+V6pnz
                                        4/7mw2fnkFsm3rHHVNn7UuFYmf8NNzNSTWvV
                                        MkPKK8dX2vHCYUmWPG9rVPr791pjXU4Yxrj/
                                        8cEHv/DzEiNuSrg5o1y0E4ashdh6tD6fwwVL
                                        mvC6iFVOp6krmN6kprLleVHU723aokhF9jLS
                                        Po6/qzdNmoPLI51rc1d+jU24PgjkJcwfi8Rh
                                        OkwEtbIz1a2zHczG/I8GqDfknpWS2X2+m1/+
                                        MSbox9Ui+k+PJtL3baRgizaz8frewJa9PU23
                                        UX2CJ31hopdgqDriNBIkInNm21K3/9wUto1D
                                        IeGEtzRgkwPV242NtQ== )
; secure
```

Imagen 16. Caché de DNS en texto plano

Para limpiar la caché, debemos usar el par de comandos:

```
rndc flush
rndc reload
```

```
root@servidor:~# rndc flush
root@servidor:~# rndc reload
server reload successful
root@servidor:~#
```

Imagen 17. Comandos para limpiar y recargar la caché de DNS

Con el comando anterior, o los comandos anteriores, hemos eliminado todas las entradas en memoria caché y, además, hemos resteado el servicio.

Un ejemplo del funcionamiento de nuestro servicio instalado lo podemos hacer mediante el comando `dig`.

El ejemplo va a funcionar del siguiente modo, preguntando a nuestra propia máquina, *localhost*, cual es la dirección de www.google.es. El comando sería el siguiente:

```
dig @localhost www.google.es
```

```
root@servidor:~# dig @localhost www.google.es

; <<>> DiG 9.16.15-Ubuntu <<>> @localhost www.google.es
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38895
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 77dab62f3fe14b3501000000062b31017e3f2d1677c7a3c2c (good)
;; QUESTION SECTION:
;www.google.es.                 IN      A

;; ANSWER SECTION:
www.google.es.          300     IN      A       142.250.201.67

;; Query time: 827 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: mié jun 22 12:50:31 UTC 2022
;; MSG SIZE  rcvd: 86
```

Imagen 18. Comando dig sin nada almacenado en caché

Podemos ver que la dirección final que nos da es 142.250.201.67, y que ha tardado 827 *msec* en procesar la operación. Vamos ahora a volver a ejecutarla:

```
root@servidor:~# dig @localhost www.google.es

; <<>> DiG 9.16.15-Ubuntu <<>> @localhost www.google.es
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25203
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: c9cbbdd78b23da900100000062b3105727d8417745a635bb (good)
;; QUESTION SECTION:
;www.google.es.                 IN      A

;; ANSWER SECTION:
www.google.es.          236     IN      A       142.250.201.67

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: mié jun 22 12:51:35 UTC 2022
;; MSG SIZE  rcvd: 86
```

Imagen 19. Comando dig con información almacenada en caché

De esta nueva imagen, comprobamos que todo sigue siendo igual, pero que ahora ha tardado 0 *msec* en ejecutar el comando, puesto que se ha almacenado la información en caché.

Igual que hemos indicado *localhost*, podríamos haber indicado un nombre o dirección de un servidor conocido, y debería de darnos la misma respuesta.

Por último, dentro de este servicio, podemos configurar nuestros propios reenviadores o *forwarders*. Esto se lleva a cabo en el fichero */etc/bind/named.conf.options*, donde en la sección *forwarders* (que en un principio aparece comentada con "*//*") indicaremos los servidores que queremos que hagan de reenviadores.

En nuestro ejemplo hemos indicado los de Google.

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {
                8.8.8.8;
                8.8.4.4;
        };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        auth-nxdomain no;
        listen-on-v6 { any; };
};
```

Imagen 20. /etc/bind/named.conf.options con forwarders editados

Para comprobar que esto realmente funciona, si volvemos a ejecutar todo el proceso anterior, desde el principio, no debe de darnos ningún error.

2.6.2. Definición de zona en el servidor y transferencia de zona

En anteriores apartados, hemos visto que hay ficheros específicos que debemos de editar para la configuración.

Vamos a explicar una configuración básica con un ejemplo: en una red nuestra interna, vamos a configurar el servidor DNS para que resuelva de manera directa e inversa y, además, que tenga una transferencia interna a otro servidor.

PARA TENER EN CUENTA...

Es muy importante saber y tener en cuenta, que puede que no todas las configuraciones que hagamos en este tema serán validas a la hora de probarlas.

Esto viene dado porque dependiendo del sistema que tengamos y su configuración de red, un pequeño cambio puede hacer que todo el sistema cambie.

Lo que en estos temas se da es un ejemplo de configuración que comprueba que la sintaxis sí que es correcta en todo momento.

Sin transferencia de zona

En nuestro ejemplo, vamos a definir la zona *universae.lan* para la red 10.0.0.0/24 teniendo como servidor principal o maestro, el servidor con IP 10.0.0.24.

El proceso sería el siguiente:

1. Lo primero que debemos de hacer es *loguearnos* como *root* en el servidor en cuestión.
2. Una vez que somos administradores, lanzamos el siguiente comando para que el *firewall* permita las entradas de peticiones DNS:

   ```
   ufw allow bind9
   ```

3. Vamos ahora a desplazarnos al directorio */etc/bind*.
4. Dentro de este directorio, hemos visto anteriormente los ficheros que tenemos, vamos a hacer primero una copia de seguridad de los archivos principales:

   ```
   cp named.conf.options named.conf.options.orig
   cp.named.conf.local named.conf.local.orig
   ```

5. Una vez que se ha realizado la copia de seguridad, pasamos a editar el primer archivo, *named.conf.options*, que debe de quedar como en la imagen siguiente:

```
acl internals {
        127.0.0.1;
        10.0.0.0/24;
};

options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {
                8.8.8.8;
                8.8.4.4;
         };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        //listen-on-v6 { any; };
};
```

Imagen 15. Configuración de fichero *named.conf.options*

En esta imagen popdemos ver que se añadido la sección *acl internals* para determinar que redes se pueden conectar al servidor de DNS.

Además, comentamos la línea *//listen-on-v6* ya que no vamos a determinar direcciones IPv6,

6. Después, nos vamos a configurar el fichero de definición de zonas, que es *named.conf.local*. En este fichero se va a tratar:

   a. La zona directa, a la que añadimos el nombre, de zona, si es maestro o esclavo, el fichero donde vamos a configurar la zona y si se pueden transferir autoridades sobre dicha zona.

   b. La zona inversa, con los mismos parámetros, como consejo, el nombre debe de ser los tres primeros octetos de la dirección de red al revés, seguidos de *.in-addr.arpa*.

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
include "/etc/bind/zones.rfc1918";

zone "universae.lan" {
        type master;
        file "/etc/bind/zones/db.universae.lan";
        allow-transfer { internals; };
};

zone "0.0.10.in-addr.arpa" {
        type master;
        file "/etc/bind/zones/db.10.0.0";
        allow-transfer { internals; };
};
```

Imagen 16. Configuración fichero *named.conf.local*

Es importante recalcar que se ha descomentando la línea *include "etc/bind/zones.rfc1918"*, que es necesaria para que funcione la zona inversa.

7. Si hemos editado los archivos, y lanzamos el comando `named-checkconf`, nos indicará que la sintaxis está bien si no muestra nada. En caso de error, nos mostrará el error.

```
root@servidor:/etc/bind# named-checkconf
root@servidor:/etc/bind#
```

Imagen 17. Comprobación de sintaxis

8. Vamos ahora a definir las zonas, para eso, dentro de este mismo directorio, lanzamos el comando:

   `mkdir zones`

   Usamos dicho comando para crear el directorio que hemos indicado en *named.conf.local* como archivo de configuración de zona.

9. Ahora, copiamos el fichero *db.local* a este directorio dos veces, uno con el nombre de la zona directa (el nombre indicado en el anterior archivo), y otro con el nombre de la zona inversa.

10. Si ya tenemos el fichero copiado, vamos a editar el de la zona directa.

11. En este fichero, añadimos los principales parámetros que se definen en la siguiente imagen, prestando especial atención a los registros que vimos en anteriores apartados y sobre todo a lo siguiente:

    El Serial o número de serie, debe de incrementarse manualmente cada vez que editemos dicho fichero, o si no, no tendrán validez los cambios por mucho que editemos.

```
;
; BIND data file for local loopback interface
;
$TTL    604800
$ORIGIN universae.lan.
@       IN      SOA     ns1.universae.lan. administrador.universae.lan. (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
universae.lan.  IN      NS      ns1.universae.lan.
ns1.universae.lan.      IN      A       10.0.0.24
www.universae.lan.      IN      CNAME   ns1.universae.lan.
```

Imagen 18. Fichero de configuración de zona directa

Podemos ver en la anterior imagen que hemos definido los registros del servidor, el registro de dirección y un registro de área.

12. Ahora, comprobamos que la zona está correcta en temas de sintaxis con el comando:

    `named-checkzone nombre_zona archivo_zona`

```
root@servidor:/etc/bind# named-checkzone universae.lan zones/db.universae.lan
zone universae.lan/IN: loaded serial 3
OK
root@servidor:/etc/bind# _
```

Imagen 19. Comprobación de zona directa

El resultado anterior nos indica que todo está correcto.

13. Vamos a hacer los dos mismos pasos anteriores con la zona inversa:

```
;
; BIND data file for local loopback interface
;
$TTL    604800
$ORIGIN 0.0.10.in-addr.arpa.
@       IN      SOA     ns1.universae.lan. administardor.universae.lan. (
                              3         ; Serial
                         604000         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604000 )       ; Negative Cache TTL
;
        IN      NS      ns1.universae.lan.
24      IN      PTR     ns1.universae.lan.
24      IN      PTR     www.universae.lan.
```

Imagen 20. Fichero de configuración de zona inversa

Hay que prestar especial atención aquí, a que los registros de definición de nombre vienen definidos indicando únicamente el octeto final, que es el que cambia con respecto a la red que hemos definido anteriormente.

14. Comprobamos ahora que la sintaxis es correcta.

```
root@servidor:/etc/bind# named-checkzone db.10.0.0.in-addr.arpa zones/db.10.0.0
zone db.10.0.0.in-addr.arpa/IN: loaded serial 5
OK
root@servidor:/etc/bind# _
```

Imagen 21. Comprobación de zona inversa

15. Reiniciamos ahora el servicio *bind9*.

16. Si ahora, definimos en otro servidor de la misma red, que este servidor sea el DNS de dichos servidores, podemos comprobar mediante el comando `ping` que funciona correctamente.

```
root@servidor2:~# ping ns1.universae.lan
PING ns1.universae.lan (10.0.0.24) 56(84) bytes of data.
64 bytes from 10.0.0.24: icmp_seq=1 ttl=64 time=0.298 ms
64 bytes from 10.0.0.24: icmp_seq=2 ttl=64 time=0.360 ms
64 bytes from 10.0.0.24: icmp_seq=3 ttl=64 time=0.348 ms
64 bytes from 10.0.0.24: icmp_seq=4 ttl=64 time=0.493 ms
64 bytes from 10.0.0.24: icmp_seq=5 ttl=64 time=0.349 ms
64 bytes from 10.0.0.24: icmp_seq=6 ttl=64 time=0.346 ms
^Z
[1]+  Stopped                 ping ns1.universae.lan
root@servidor2:~#
```

Imagen 22. Ping

Con transferencia de zona

Realizamos todas las configuraciones previas en el servidor que va a funcionar como esclavo para que sea exactamente igual que el maestro.

Ahora, de nuevo en el maestro, definimos los ficheros del siguiente modo:

1. El fichero *named.conf.local:*

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
include "/etc/bind/zones.rfc1910";

zone "universae.lan" {
        type master;
        file "/etc/bind/zones/db.universae.lan";
        allow-transfer { 10.0.0.25; };
        notify yes;
        allow-query {any; };
};

zone "0.0.10.in-addr.arpa" {
        type master;
        file "etc/bind/zones/db.10.0.0";
        allow-transfer { 10.0.0.25; };
};
```

Imagen 23. *named.conf.local* con nueva configuración de transferencia

En la zona directa e inversa, indicamos que se pueda transferir la zona al servidor con IP 10.0.0.25.

Además, en la zona directa únicamente, añadimos notificaciones y permiso para solicitud a cualquiera.

2. Ahora editamos los ficheros de ambas zonas añadiendo el segundo servidor:

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.universae.lan. admin.universae.lan. (
                              9         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.universae.lan.
ns1     IN      A       10.0.0.24
www     IN      CNAME   universae.lan.
@       IN      NS      ns2.universae.lan.
ns2     IN      A       10.0.0.25
```

Imagen 24. Fichero de configuración de zona directa modificado

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.universae.lan. admin.universae.lan. (
                              10            ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.universae.lan.
@       IN      NS      ns2.universae.lan.
24      IN      PTR     ns1.universae.lan.
25      IN      PTR     ns2.universae.lan.
```

Imagen 25. Fichero de configuración de zona inversa modificado

Nos vamos en este momento al servidor esclavo y configuramos el fichero *named.conf.local* del siguiente modo:

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
include "/etc/bind/zones.rfc1918";

zone "universae.lan" {
        type slave;
        masters {10.0.0.24; };
        allow-query {any; };
        file "/var/cache/bind/db.transferuniversae";
};

zone "0.0.10.in-addr.arpa" {
        type slave;
        masters {10.0.0.24; };
        allow-query {any; };
        file "/var/cache/bind/db.10.0.0trasnfer";
};
```

Imagen 26. Fichero *named.conf.local* en servidor esclavo

En ambas zonas hemos descrito que el tipo es esclavo, la IP del servidor principal como maestro, que se admitan todas las consultas, y el fichero de configuración.

Podemos ver, que el fichero de configuración en sí tiene una dirección distinta a la anterior.

Comprobamos la sintaxis del servicio y acto seguido, nos dirigimos a la ruta */var/cache/bind*. En esta ruta, vemos que estos archivos no están.

Acto seguido debemos de reiniciar el servicio de DNS en ambos servidores, entrar de nuevo en dicha ruta y ver que se han copiado los archivos de configuración de zonas del servidor maestro al servidor esclavo, pero cambiando el nombre por el nuestro indicado.

Nuestro DNS ya estaría configurado para dar un servicio interno completo.

2.7. Servidores de nombres en direcciones IP dinámicas

Hemos visto anteriormente como funciona el servidor de DNS como un elemento asilado y estáticos, es decir, que las direcciones IP son estáticas, predefinidas y por tanto, los nombres de dominio van asociados a una dirección IP estática también, pero esto no siempre es así.

Como sabemos, existe el servicio DHCP, y muchas veces, este coexiste con un servidor de DNS, por lo que es necesario actualizar en muchas ocasiones los registros con nuevos valores dependiendo de la IP que se tenga en dicho momento.

En esto consiste el DDNS o DNS dinámico, en mantener de manera constante los nombres de dominio actualizados.

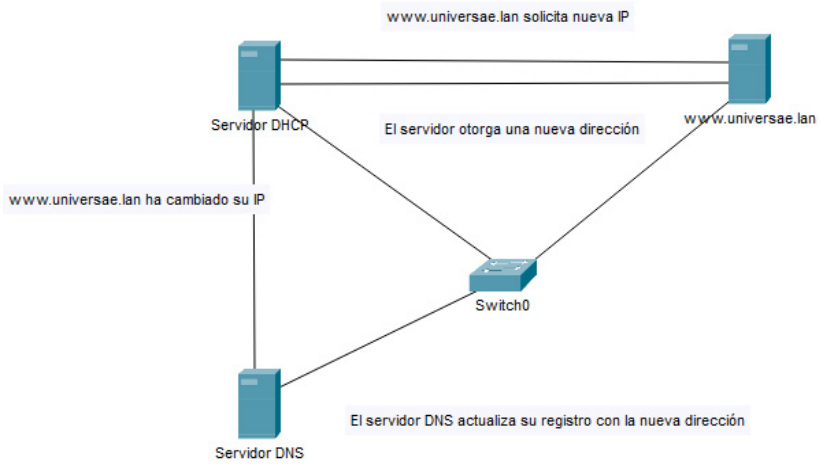

Imagen 25. Funcionamiento de un servicio DDNS

Como vemos en la imagen anterior, el servidor DHCP es el encargado de indicar el cambio de IP al DNS y este último actualiza sus registros.

Por último, saber que el servicio DDNS se puede proporcionar en dos niveles distintos:

- **Nivel externo:** el servidor de DDNS se aloja en internet y se usa cuando se necesita de sincronización con el servidor DHCP proporcionado por el ISP con el nombre de dominio contratado.
- **Nivel interno:** el servidor DDNS se encuentra en nuestra red local ye s usado por equipos internos que siempre deben estar en constante actualización con el servidor DHCP interno.

Este tipo de tecnología no es común encontrárnosla en redes internas por muy grandes que sean, sino en servidores online que alojen sitios web o servidores públicos web, donde el cambio de IP es común.

www.universae.com