Servicios avanzados I: Servidor de nombres (DNS).

Autor: Enrique V. Bonet Esteban

Introducción.

En Internet, los paquetes de datos viajan mediante el protocolo de red IP. Dicho protocolo, ya sea en su versión 4 o en su versión 6, funciona mediante lo que se conoce como direcciones IP, esto es, conjunto de números que especifican de forma unívoca el ordenador al que deseamos enviar el paquete de datos.

Sin embargo y de forma general, todos estamos acostumbrados a utilizar nombres para identificar a los ordenadores (*www.uv.es*, *irtic.uv.es*, *www.google.es*, etc.). Estos nombres, sencillos de recordar, deben ser convertidos previamente en direcciones IP para poder conocer el ordenador destino de los paquetes de datos que deseamos enviar. Para cumplir con esta función se desarrolló el servicio de nombres de dominio (Domain Name Server). El DNS es un servicio que permite convertir el nombre de cualquier ordenador en su dirección IP y viceversa.

El precedente del DNS se encuentra en la época en que Internet todavía no había alcanzado su actual grado de expansión y se conocía todavía bajo el nombre de ARPANET. En aquella época existía un solo fichero, de nombre *host.txt*, que administraba el Centro de Información de Red del Departamento de Defensa de EE.UU. (Departament Of Defense Network Information Center), y que contenía todas las relaciones entre los nombres de los ordenadores y sus direcciones IP. Dicho fichero era editado en un solo ordenador y distribuido, de forma periódica, a todos los ordenadores que formaban ARPANET mediante una conexión utilizando el protocolo FTP.

Sin embargo, con el crecimiento del número de ordenadores conectados a ARPANET y su evolución hasta lo que hoy conocemos como Internet, lo que ha conllevado un espectacular aumento del número de ordenadores conectados a la red, la existencia de un solo fichero central que se distribuya es impracticable, tanto por el tamaño del mismo, como por la gran cantidad de nuevas relaciones entre nombres y direcciones IP que sería necesario insertar de forma diaria en dicho fichero.

Debido a ello, surgió la necesidad de crear un servicio que permitiera que dicho fichero *host.txt* se dividiera en ficheros distribuidos, de forma que cada entidad que poseyera un dominio en Internet se encargara de manejar y actualizar su propio fichero, fichero que contendría únicamente las relaciones entre los nombres y las direcciones IP de los ordenadores de su dominio. Como último paso, se debería crear un fichero central que contendría la relación entre los dominios de Internet y las direcciones IP de los ordenadores que podían suministrar la información sobre las direcciones IP y los nombres de los ordenadores existentes en un dominio de Internet.

Aunque en una primera impresión pueda parecer que todo el trabajo realizado en el desarrollo del DNS se limita a suplir los problemas de memoria del usuario, capaz de recordar nombres pero no secuencias numéricas, y que bastaría con que el usuario en lugar de poner http://www.uv.es pusiera http://147.156.1.4 para evitar la necesidad del DNS, esa primera impresión es errónea. Basta, por ejemplo, con citar la existencia de

servidores virtuales (Virtual Host) de páginas Web[1]. Estos servidores virtuales se basan en la existencia de un único servidor Web, que se encuentra en la misma dirección IP, y que proporciona una u otra página Web en función del nombre del "ordenador virtual" que se solicite mediante su nombre. Un ejemplo de ello podemos verlo en las tres siguientes páginas: *http://irtic.uv.es*, *http://autismo.uv.es* y *http://www.cdlibre.org*. En todos los casos el servidor se encuentra en la misma dirección IP (147.156.222.65), pero según se le llame con uno u otro nombre, proporciona páginas Web distintas.

Existen un gran número de clientes de DNS, en concreto, cualquier ordenador conectado a Internet es un cliente potencial de un DNS, mientras que solo un conjunto reducido de ordenadores conectados a Internet es un servidor de DNS en Internet. Además, todo ordenador servidor de DNS es a su vez un cliente de DNS, pues debe usar su servicio de DNS para resolver los nombres que él mismo no es capaz de resolver por pertenecer a otro dominio de Internet.

El cliente DNS.

La existencia de un cliente DNS es necesaria en todo ordenador conectado a Internet, excepto que deseemos realizar cualquier operación en Internet conociendo la dirección IP de los ordenadores y no su nombre, y además deseemos no utilizar servicios como el de los servidores virtuales de páginas Web.

El funcionamiento del cliente DNS se basa en tres ficheros, *hosts*, *host.conf* y *resolv.conf*, que pasamos a explicar de forma detallada a continuación.

El primero de estos ficheros, */etc/hosts*[2], puede considerarse como una herencia del antiguo fichero *host.txt* de ARPANET que hemos comentado con anterioridad. El fichero contiene la relación entre los nombres y las direcciones IP de todos los ordenadores necesarios en el arranque del ordenador, cuando este todavía no tiene activados sus interfaces de acceso a la red y por tanto no puede consultar los servidores de DNS. Aunque dicho fichero esta formado por dos líneas, pueden añadirse en él todas las líneas que se deseen, conteniendo cada una la relación entre los nombres de un ordenador y su dirección IP. Un ejemplo del fichero es el siguiente:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1       localhost localhost.localdomain localhost6 localhost6.localdomain6
147.156.222.65 glup.uv.es glup.irobot.uv.es glup
```

La primera línea, 127.0.0.1 es siempre necesaria, pues hace referencia a *localhost*, el interfaz de loopback del ordenador, y es necesaria en el arranque del ordenador, antes de activar sus dispositivos de acceso a la red, mientras que la segunda línea es su equivalente para IPv6.

El segundo de estos ficheros, */etc/host.conf*, indica, entre otros valores, el orden en que deben utilizarse las posibilidades disponibles para la resolución de los nombres, esto es, si primero debe mirarse el fichero */etc/hosts* y en caso de no encontrar allí la resolución del nombre debe consultarse al servidor de nombres o bien debe hacerse al contrario. La entrada típica de este fichero es:

[1] En temas posteriores estudiaremos el servidor web Apache y la configuración de servidores virtuales.
[2] En Windows el fichero */etc/hosts* se encuentra en %systemroot%*system32**drivers**etc**hosts.*

```
order hosts,bind
```

Que indica que el orden es primero mirar el fichero local (*hosts*) y, en caso de no obtener respuesta, recurrir al servidor de nombres (*bind*)[3].

La configuración del fichero */etc/host.conf* de esta forma permite minimizar las consultas a los servidores DNS, pues es posible introducir en el fichero /etc/hosts las direcciones IP de aquellos ordenadores que son utilizados con más frecuencia, y además permite "falsificar" las direcciones IP de los ordenadores que deseemos, pudiendo asignar al nombre de un ordenador la dirección IP que deseemos[4]. Otra entrada que suele aparecer en el fichero /etc/hosts es la entrada:

```
multi on
```

Que indica que en caso de que un nombre se corresponda con más de una dirección IP en el fichero */etc/hosts* se devuelvan todas las entradas en lugar de solo la primera.

El tercer fichero, */etc/resolv.conf*, indica el dominio al que pertenece este ordenador y las direcciones IP de los servidores de nombres a los que pueden efectuarse las consultas. Los parámetros de configuración más utilizados en este fichero son *domain*, *search* y *nameserver*.

El parámetro *domain* indica el dominio al que pertenece el ordenador, de forma que se puedan realizar consultas de nombres de ordenadores pertenecientes al dominio sin que sea necesario escribir el nombre completo del ordenador, esto es, su nombre más el nombre del dominio. Así, la especificación con el parámetro *domain* del dominio *uv.es*, permite realizar una consulta de nombre de ordenador "post" y obtener la respuesta 147.156.0.253, sin necesidad de preguntar por su nombre completo, esto es "post.uv.es".

El segundo parámetro, *search*, es una generalización del anterior, de forma que es posible especificar hasta seis dominios donde la suma de la longitud de los nombres de todos ellos no exceda de 256 caracteres. Este parámetro, a pesar de representar una ventaja sobre el anterior, no suele usarse, o bien se usa con un solo dominio, de forma similar a *domain*, pues en caso contrario obliga a los servidores a generar una gran cantidad de tráfico, realizando consultas a los servidores de todos los dominios que se especifiquen hasta encontrar la respuesta, si esta existe. Como consecuencia añadida del efecto anterior, la resolución de nombres puede volverse muy lenta, llegando los clientes a producir fallos por suceder un "timeout" en la espera de la respuesta.

El último parámetro, *nameserver*, indica las direcciones IP de los servidores de red que podemos consultar. Pueden existir tantas entradas *nameserver* como se deseen, aunque solo se utilizan las tres primeras, consultándose los servidores en el orden en que se encuentran sus entradas. Por ello, deben ponerse en primer lugar aquellos servidores que se consideren más rápidos y disponibles, pues ello disminuye el tiempo de respuesta, tanto por la rapidez del servidor, como por el hecho de que si un servidor

[3] Esta entrada es la que se utiliza por defecto si no se especifica la opción *order* en el fichero.

[4] Esto puede ser útil, por ejemplo, en la configuración de aplicaciones que se conecten a un servidor que este en producción, pues podemos asignar el nombre del servidor con la dirección de otro servidor que no este en producción y testear la aplicación contra el mismo.

esta disponible de forma intermitente en el tiempo, toda pregunta que se le realice cuando no esta disponible deberá esperar el que suceda un "timeout" para enviar dicha pregunta al siguiente servidor de nombres.

Un ejemplo de fichero */etc/resolv.conf* es el siguiente:

```
domain          uv.es
nameserver      147.156.222.65
nameserver      147.156.1.1
nameserver      147.156.1.3
```

Consulta de un cliente DNS a un servidor DNS.

Generalmente, toda consulta de un cliente DNS a su servidor suele realizarla el programa que invocamos (telnet, ftp, correo, navegador web, etc.). Sin embargo, existe un programa que nos permite realizar consultas interactivas como cliente DNS. Dicho programa es */usr/bin/host*[5].

Así, si deseamos consultar la dirección IP del ordenador "glup" del Instituto de Robótica de la Universitat de València, basta con ejecutar:

```
host glup.irobot.uv.es
```

Y obtener como respuesta la siguiente línea.

```
glup.irobot.uv.es has address 147.156.222.65
```

De igual forma, podemos realizar una consulta inversa, esto es, introducir la dirección IP de un ordenador y obtener su nombre. Por ejemplo:

```
host 147.156.222.65
```

Devuelve como respuesta la línea:

```
65.222.156.147.in-addr.arpa domain name pointer glup.irobot.uv.es
```

Si se observa, se obtiene una respuesta donde la dirección IP viene dada en formato inverso, esto es así pues toda consulta inversa debería ponerse en el formato inverso de la dirección IP, pues si se observa, al preguntar por "glup.irobot.uv.es", estamos poniendo la pregunta como ordenador, dominio terciario, dominio secundario, dominio primario, formato que se conservaría si ponemos la dirección IP del ordenador como 65.222.156.147 y no si la ponemos al revés, tal y como solemos efectuar normalmente. De todas formas, y como hemos podido comprobar, el programa admite direcciones puestas en el formato en que son usadas normalmente por nosotros y traducidas automáticamente al formato correcto.

Por otro lado, la introducción del dominio en el fichero */etc/resolv.conf*, explicada con anterioridad, nos permite que el ordenador realice consultas donde solo se

[5] En sistemas operativos UNIX y Windows, el programa que permite realizar las consultas recibe el nombre de nslookup. Este programa se encuentra también disponible en las versiones actuales de Linux.

especifique el nombre del ordenador y el dominio se añada de forma automática, como puede verse en el ejemplo siguiente[6]:

```
host glup
```

Devolviéndonos la respuesta:

```
glup.uv.es is an alias for glup.irobot.uv.es
glup.irobot.uv.es has address 147.156.222.65
```

Donde se nos indica que "glup.uv.es" es un alias del ordenador "glup.irobot.uv.es", devolviendo la dirección de este último ordenador[7].

Si en cualquier consulta no se desea que el dominio sea añadido, basta con poner un punto al final de la consulta. Así, la consulta siguiente, que solo se diferencia en el punto puesto al final, proporciona un mensaje de que el ordenador no existe en la red.

```
host glup.
Host glup not found. 3 (NXDOMAIN)
```

Indicándonos que el ordenador "glup" no existe, pues hemos indicado explícitamente que no deseamos que añada el dominio en la consulta.

El comando *host* posee múltiples opciones, pudiendo obtenerse un listado de las mismas en las páginas del manual del sistema, siendo de destacar la opción *-t type*. Esta opción permite seleccionar el tipo de registro cuya respuesta queremos obtener[8]. Así, por ejemplo, si ejecutamos el comando:

```
host -t A mirror.uv.es
```

Obtenemos como respuesta:

```
mirror.uv.es has address 147.156.223.157
```

Que es la respuesta que se obtiene por defecto, mientras que si ejecutamos el comando:

```
host -t AAAA mirror.uv.es
```

Obtenemos como respuesta:

```
mirror.uv.es has AAAA address 2001:720:1014:222::2
```

Y si ejecutamos el comando:

[6] Esto puede realizarse no solo con el comando host, sino también con cualquier otro comando como telnet, ftp, etc., por lo cual puede solicitarse, por ejemplo, la conexión al host glup como telnet glup.

[7] El hecho de que funcione la consulta a glup.uv.es es debido a que la Universitat de València no permite que dos ordenadores de su dominio secundario uv.es tengan el mismo nombre, lo que posibilita la existencia de un alias entre los nombres completos de todos los ordenadores de la Universitat de València y sus nombres abreviados al dominio secundario.

[8] Los tipos de registro existentes se verán con posterioridad en este tema.

```
host -t MX mirror.uv.es
```

Obtenemos como respuesta:

```
mirror.uv.es mail is handled by 20 postin.uv.es
```

El servidor DNS.

Antes de introducirnos en la explicación de un servidor DNS, es necesario resaltar que toda organización que desee tener un dominio propio debe, como mínimo, tener registrados dos servidores de DNS en la autoridad de registro de nombres, esto es, en InterNIC, la cual añade esos dos servidores a la información que contienen los servidores raíz[9], indicando que esos dos servidores serán los encargados de realizar la resolución de nombres y direcciones en ese dominio.

Los servidores de DNS se clasifican en cuatro conjuntos no disjuntos:

- Servidor primario, que es el servidor que posee los ficheros sobre los que se introducen las modificaciones del dominio debidas a las altas, bajas, modificaciones, etc., producidas en los ordenadores del dominio.

- Servidor secundario, que es cualquier otro servidor del dominio que no posee los ficheros del dominio y que, por tanto, debe obtener de otro servidor los mismos.

- Servidor maestro, que es un servidor, primario o secundario, que permite que otros ordenadores accedan al mismo y obtengan los ficheros con la información del dominio. Este proceso es conocido como transferencia del fichero de zona o dominio. En todo dominio, el servidor primario debe ser un servidor maestro, pudiendo existir servidores secundarios que actúen también como servidores maestros.

- Servidor esclavo, que es un servidor que obtiene los ficheros del dominio y debe obtener los mismos de otro servidor. Todos los servidores, excepto el servidor primario, son servidores esclavos de, como mínimo, un servidor maestro.

Entrando ya en aspectos más técnicos, el servidor DNS se basa en la ejecución del demonio */usr/sbin/named*. El servidor de DNS utiliza el puerto 53 UDP para la recepción de consultas y el envío de sus respuestas. Además, suele utilizar, en caso necesario, el puerto 53 TCP para las transferencias del fichero de zona.

El demonio *named* posee multitud de parámetros configurables por línea de comandos, tales como el nombre del fichero de configuración, el puerto UDP que va a utilizar para efectuar su cometido, etc. Sin embargo, las únicas opciones que suelen utilizarse con las opciones -u y -t, tal y como puede verse en la información obtenida mirando la ejecución del demonio en un ordenador[10]:

[9] Podemos comparar estos servidores raíz con el índice de un libro, que es una referencia que indica donde podemos encontrar la información que buscamos.

[10] Una forma de ver dicha información es ejecutar la siguiente sentencia en la línea de comandos de la shell del sistema: ps -ef|grep named.

```
named    2383    1  0 Sep09 ?     00:19:41 /usr/sbin/named -u
named -t /var/named/chroot
```

La opción -u indica el usuario con el que se ejecutará el demonio *named* una vez termine su inicialización. Esta es una medida de seguridad, pues el demonio es ejecutado por el administrador del sistema (usuario root), pudiendo, mediante este cambio, limitar los privilegios de dicho demonio, una vez inicializado, a los que posea el usuario indicado en dicha opción. En la actualidad, Linux suele tener configurado por defecto un usuario llamado igual que el demonio, esto es "named" y que posee solo permisos para escribir en los ficheros y directorios necesarios para ejecutar su función.

Por su parte, la opción -t, que debe usarse en conjunción con la anterior, indica el directorio donde será "encerrado" el demonio en su ejecución, aumentando la seguridad del mismo, pues no se le permite acceder a los archivos, etc., situados fuera del directorio especificado.

Tipos de registros.

La configuración de todo DNS se basa en una serie de palabras clave llamadas registros. Estos registros indican el tipo de datos que lo acompañan y a que corresponden, por ejemplo, una resolución del nombre del ordenador a su dirección IP, una resolución de la dirección IP de un ordenador a su nombre, etc. Los principales tipos de registros son los siguientes[11]:

- SOA (Start Of Autority). Especifica información de la autoridad de un dominio DNS, incluyendo el nombre del servidor primario del dominio, el correo del administrador del dominio, el número de serie del dominio y parámetros temporales relacionados con los tiempos de refresco del dominio.

- NS (Name Server). Permite delegar las respuestas de información de un dominio en el ordenador especificado en esta entrada.

- PTR (Pointer To Register). Hace corresponder una dirección IPv4 o IPv6 con el nombre del ordenador. Estas entradas son usadas para la resolución inversa (de dirección IP a nombre).

- A (Address). Hace corresponder un nombre de ordenador con una dirección IPv4. Estas entradas son usadas para la resolución directa (de nombre a dirección IPv4).

- AAAA (Address). Hace corresponder un nombre de ordenador con una dirección IPv6. Estas entradas son usadas para la resolución directa (de nombre a dirección IPv6).

- CNAME (Canonical NAME). Indica un nombre alternativo (alias) de un ordenador registrado y que por tanto tiene su correspondiente entrada A o AAAA.

[11] Existen muchos más tipos de registro de los aquí especificados, siendo los aquí especificados los de uso más común en Internet.

- MX (Mail eXchanger). Indica los ordenadores que aceptan el correo destinado al ordenador que se indique en la entrada.

Configuración de un servidor de DNS.

La configuración del servidor de DNS se realiza en dos ficheros, */etc/sysconfig/named* y en el fichero */etc/named.conf* o fichero similar si la ejecución del servidor es encerrada en un directorio.

El fichero */etc/sysconfig/named* contiene las opciones que se desean pasar al demonio de named en su arranque. En versiones anteriores contenía de forma general las opciones -u y -t que se han comentando con anterioridad. Sin embargo, con posterioridad la opción -u fue incluida por defecto en el script de arranque de Fedora y la opción -t para "encerrar" el demonio ha sido incluida en la actualidad en uno de los scripts de arranque, de forma que:

```
systemctl start named.service
```

Arranca el servidor de DNS sin encerrarlo en un directorio, mientras que:

```
systemctl start named-chroot.service
```

Arranca el servidor de DNS encerrándolo en el directorio especificado, que por defecto es */var/named/chroot*[12].

El otro fichero de configuración del demonio *named* es */etc/named.conf*. Debe tenerse en cuenta que si el servicio es encerrado en el directorio */var/named/chroot*, el fichero será */var/named/chroot/etc/named.conf*. De todas formas, el actual script de arranque *named-chroot.service* copia los ficheros necesarios de */etc* en el directorio donde el demonio named es encerrado, por lo que puede modificarse el fichero */etc/named.conf* y antes de arrancar el servidor los ficheros serán copiados a la localización correcta.

Actualmente, el fichero */etc/named.conf* contiene las siguientes líneas:

```
options {
      listen-on port 53 { 127.0.0.1; };
      listen-on-v6 port 53 { ::1; };
      directory  "/var/named";
      dump-file  "/var/named/data/cache_dump.db";
      statistics-file "/var/named/data/named_stats.txt";
      memstatistics-file "/var/named/data/named_mem_stats.txt";
      allow-query     { localhost; };
      recursion yes;

      dnssec-enable yes;
      dnssec-validation yes;
      dnssec-lookaside auto;

      /* Path to ISC DLV key */
```

[12] Por seguridad debería siempre arrancarse la opción que es encerrada en un directorio para su ejecución.

```
      bindkeys-file "/etc/named.iscdlv.key";

      managed-keys-directory "/var/named/dynamic";

      pid-file "/run/named/named.pid";
};

logging {
      channel default_debug {
            file "data/named.run";
            severity dynamic;
         };
};

zone "." IN {
      type hint;
      file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

La palabra clave *options* especifica la sección dentro de la cual se especifican las opciones de configuración global del demonio. El conjunto de opciones de configuración es muy numeroso, por lo que vamos a centrarnos unicamente en las que aparecen en el fichero de configuración que viene por defecto en Fedora.

Las opciones *listen-on* y *listen-on-v6* especifican el puerto y los interfaces de red en los que permanece a la escucha el servidor. Como puede verse en la configuración por defecto, el puerto que utiliza es el puerto 53 UDP[13], y los interfaces de red son únicamente los de loopback en versiones IPv4 e IPv6, por lo que si se desea que otros ordenadores puedan consultar nuestro servidor de DNS deberíamos modificar el valor de dichas opciones.

La opción *directory* especifica el directorio donde se ejecutara *named*. Debe tenerse en cuenta que si *named* ha sido encerrado en algún directorio, el camino aquí especificado será relativo al directorio donde se encuentre encerrado *named* en su ejecución.

Las opciones *dump-file*, *statistics-file* y *memstatistics-file* especifican los ficheros donde deberán almacenarse los datos de la cache de DNS, las estadísticas de uso del servidor de DNS y las estadísticas de uso de memoria por parte del servidor de DNS respectivamente cuando se solicite su volcado por parte del programa *rndc*[14].

Por otra parte, la opción *allow-query* especifica una lista de direcciones IP que pueden enviar consultas a este servidor. Si no se especifica esta opción, todos los ordenadores están autorizados a enviar consultas.

La opción *recursion* puede tomar los valores *yes* (valor por defecto) o *no*, e indica si el servidor podrá realizar consultas recursivas o no. Una consulta recursiva es

[13] El puerto 53 UDP es el puerto por defecto asignado para los servidores de DNS.
[14] Con posterioridad veremos el funcionamiento del programa rndc.

necesaria, por ejemplo, cuando al consultar el nombre de un ordenador se obtiene como respuesta que es un registro de tipo CNAME (alias), si se desea devolver su dirección IP, es necesario consultar el nombre del ordenador del que es alias la consulta.

Las opciones *dnssec-enable*, *dnssec-validation* y *dnssec-lookaside*, junto con la opción *bindkeys-file* y *managed-keys-directory* son opciones de configuración de la validación de seguridad del servidor DNS (DNSSEC)[15], y no entraremos en su exposición, indicando simplemente que pueden comentarse si se desea o bien dejarse con los valores por defecto para un correcto funcionamiento del servidor DNS[16].

Por último, la opción *pid-file* indica el fichero donde named almacenará su identificador de proceso (PID). Si esta entrada no existe el identificador se almacenará en la ubicación por defecto del sistema operativo, en nuestro caso */var/run*.

Después de la sección *options*, la sección *logging* indica el tipo de log, fichero, etc., donde se almacenara la información de log del servidor. Si no se especifica esta entrada, el log se guardará en el fichero por defecto del sistema, generalmente */var/log/messages*. Igual que en las opciones de seguridad, comentar que puede comentarse si se desea o dejar los valores por defecto para el correcto funcionamiento del servidor DNS[17].

A continuación de la sección *logging* se encuentran la secciones que definen las zona o zonas de Internet de las que somos servidores de nombres. Las zonas se definen con la sintaxis:

```
zone "<nombre de la zona>" IN {
     type <tipo>;
     file "fichero";
     allow-update { <direcciones IP>; };
     masters { <direcciones IP>; };
};
```

Donde <*tipo*> es el tipo de servidor, y puede tomar los valores *hint*, *master* o *slave*. El valor *hint* indica un servidor raíz de Internet, *master* que es un servidor maestro y *slave* que es un servidor esclavo y que por tanto obtiene los datos de un servidor maestro.

El valor *file* indica el fichero donde se guardan los datos de la zona de la que se es servidor, sea del tipo que sea.

La entrada *allow-update* indica las direcciones IP que pueden enviar actualizaciones dinámicas de los datos de esta zona, por ejemplo, servidores DHCP[18] que modifican de forma dinámica las direcciones IP de los equipos. Puede especificarse

[15] La validación de seguridad de los servidores DNS, conocido como DNSSEC, introduce una identificación de seguridad de los DNS, de forma que se tenga la certeza de que el DNS que responde las consultas es el correcto y no un DNS intruso que puede falsificar las respuestas para enviarnos, por ejemplo, a una página web incorrecta.

[16] Puede encontrarse una información detallada de que es DNSSEC y los parámetros de configuración en la URL http://www.nlnetlabs.nl/publications/dnssec_howto/

[17] Puede encontrarse una información más detallada de la sección logging en la URL http://zytrax.com/books/dns/ch7/logging.html.

[18] En temas posteriores veremos los servidores de DHCP.

el valor *none* para indicar que ninguna dirección IP esta autorizada a modificar de forma dinámica los datos de la zona.

Por último, *masters* indica en los servidores esclavos (tipo *slave*) las direcciones IP de los servidores maestros de los que el servidor esclavo puede descargarse los ficheros de datos mediante el proceso conocido como transferencia de zona. Obviamente, todo servidor esclavo debe tener una entrada *masters* con una dirección IP como mínimo.

En nuestro fichero de ejemplo podemos ver definida una zona, cuyo nombre es el signo de puntuación punto, y una línea a continuación:

```
include "/etc/named.rfc1912.zones";
```

Esta línea incluye el ficheros indicado, el cual contiene las zonas mínimas (junto con la zona punto) que debe tener configuradas como mínimo todo servidor de DNS. Estas zonas se refieren a zonas que obligatoriamente debe resolver el DNS, como es la zona 1.0.0.127.in-addr.arpa, que corresponde al interfaz de loopback del ordenador[19].

La zona ".", contiene la información sobre los servidores raíz de DNS, esto es, aquellos servidores que contienen la información sobre que ordenadores son los servidores de cada dominio en Internet. El fichero que contiene dicha información puede obtenerse mediante la ejecución de ftp como usuario anónimo en el ordenador *ftp.internic.net*. El fichero que contiene la información es *db.cache* y se encuentra como */domain/db.cache* en el servidor. La especificación de la zona "." se realiza con la siguientes líneas:

```
zone "." IN {
      type hint;
      file "named.ca";
};
```

Donde el fichero *named.ca* es el obtenido de *ftp.internic.net*[20]. Dicho fichero puede ser renombrado a cualquier otro nombre, debiendo en ese caso poner en la entrada *file* el nombre correspondiente.

Las entradas del fichero *db.cache* (o *named.ca* en nuestro caso) tienen todas la misma estructura[21]:

```
.                          518400     NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.        3600000    A     198.41.0.4
A.ROOT-SERVERS.NET.        3600000    AAAA  2001:503:BA3E::2:30
```

Donde el . inicial indica que es un servidor raíz, el 518400 o el 3600000 indica la clase de registro en Internet, A.ROOT-SERVERS.NET indica el nombre del servidor raíz y 198.41.0.4 y 2001:503:BA3E::2:30 son la direcciones IPv4 e IPv6

[19] Información sobre las zonas obligatorias, etc., puede obtenerse en el RFC 1912.

[20] Ponemos aquí el nombre *named.ca* debido a que es el nombre que posee por defecto el fichero distribuido con las versiones de Fedora. Obviamente ese nombre puede modificarse por el que se desee cambiando tanto el nombre del fichero como su valor en la configuración del servidor.

[21] En algunas entradas no aparece la línea del tipo de registro IPv6 pues ese servidor aún no puede ser consultado utilizando IPv6.

respectivamente del servidor raíz. Actualmente, el fichero contiene entradas desde la A hasta la M, lo cual indica la existencia de 13 servidores raíz en toda la red Internet[22].

Las zonas obligatorias que se encuentran incluidas son las que permiten traducir el interfaz de loopback en su nombre localhost y viceversa. Estas zonas obligatorias se introducen en el fichero de configuración de *named* como:

```
zone "localhost.localdomain" IN {
     type master;
     file "named.localhost";
     allow-update { none; };
};

zone "localhost" IN {
     type master;
     file "named.localhost";
     allow-update { none; };
};

zone
"1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
.ip6.arpa" IN {
     type master;
     file "named.loopback";
     allow-update { none; };
};

zone "1.0.0.127.in-addr.arpa" IN {
     type master;
     file "named.loopback";
     allow-update { none; };
};

zone "0.in-addr.arpa" IN {
     type master;
     file "named.empty";
     allow-update { none; };
};
```

En estas entradas se indica que el servidor es un servidor primario (t*ype master*) de los dominios “localhost.localdomain”, “localhost”, “1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa”, “1.0.0.127.in-addr.arpa” y “0.in-addr.arpa” y que la información sobre esos dos dominios se encuentra en los ficheros de nombre “named.localhost”, “named.loopback” y “named.empty”, según lo indicado en cada una de las línea *file*. Además, se indica que los ficheros no pueden ser actualizados por ningún otro ordenador mediante la línea *allow-update { none; };*.

El contenido del fichero “named.localhost” es el siguiente:

```
$TTL 1D
```

[22] Si se consulta el fichero original se puede ver el orden original en que se encuentran las líneas. Aquí se han agrupado para una mejor explicación.

```
@     IN SOA     @ rname.invalid. (
                                0     ; Número de serie.
                                1D    ; Tiempo de validez.
                                1H    ; Tiempo de reintento.
                                1W    ; Tiempo de expiracion.
                                3H    ; Validez de las consultas.
                                )
      NS    @
      A     127.0.0.1
      AAAA  ::1
```

Mientras que el contenido del fichero “named.loopback” es el siguiente:

```
$TTL 1D
@     IN SOA     @ rname.invalid. (
                                0     ; Número de serie.
                                1D    ; Tiempo de validez.
                                1H    ; Tiempo de reintento.
                                1W    ; Tiempo de expiracion.
                                3H    ; Validez de las consultas.
                                )
      NS    @
      A     127.0.0.1
      AAAA  ::1
      PTR   localhost.
```

Y el contenido del fichero “named.empty” es:

```
$TTL 3H
@     IN SOA     @ rname.invalid. (
                                0     ; Número de serie.
                                1D    ; Tiempo de validez.
                                1H    ; Tiempo de reintento.
                                1W    ; Tiempo de expiracion.
                                3H    ; Validez de las consultas.
                                )
      NS    @
      A     127.0.0.1
      AAAA  ::1
```

Donde el símbolo @ representa el dominio del que se es servidor primario y podría sustituirse por “localhost.localdomain”, “localhost”, “1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa”, “1.0.0.127.in-addr.arpa” y “0.in-addr.arpa” en cada uno de los casos sin ningún problema[23].

Los valores que se especifican dentro del tipo de registro SOA corresponden al número de serie[24], cada cuanto tiempo deben refrescar los servidores esclavos los datos, en caso de error en el refresco de los datos cuanto tiempo hay que dejar pasar entre reintentos de refresco, el tiempo de validez de los datos, transcurrido el cual sin posibilidad de refresco el servidor de DNS se considerará invalido, y el tiempo mínimo en que el servidor considera validas las consultas a otros servidores de DNS y las

[23] Puede comprobarse como el utilizar el carácter @ permite que dos ficheros se utilicen con el mismo contenido para dos dominios distintos, por ejemplo localhost.localdomain y localhost.
[24] Con posterioridad veremos con más detalle la importancia del número de serie.

almacenará en su cache para responderlas sin necesidad de consultar otra vez a otros servidores de DNS.

Por último, aparece en la configuración la línea:

```
include "/etc/named.root.key";
```

Que indica que se incluya el citado fichero que contiene la clave DNSSEC de la zona ".".

Servidor primario de DNS.

Hasta ahora solo tenemos introducidas la zona raíz y las zonas obligatorias, que no proporcionan ninguna resolución de nombres para un dominio excepto el dominio del interfaz de loopback.

Si deseamos configurar un servidor primario de un dominio, por ejemplo, el dominio "uv.es", debemos añadir las zonas que especifican los ficheros que resuelven los nombres y direcciones IP, tanto IPv4 como IPv6 de nuestro dominio. Estas zonas podemos añadirlas directamente en el fichero */etc/named.conf*, o bien en un nuevo fichero e incluir ese fichero en */etc/named.conf* tal y como hemos visto que se realiza con las zonas mínimas que debe tener configurado el servidor (línea *include*), siendo esta segunda opción más recomendable pues se modifica de forma mínima el fichero */etc/named.conf*. Estas entradas serían, para el caso del dominio "uv.es":

```
zone "156.147.in-addr.arpa" IN {
      type master;
      file "master.147.156";
      allow-update{ none; };
};

zone "4.1.0.1.0.2.7.0.1.0.0.2.ip6.arpa" IN {
      type master;
      file "master.2001:720:1014";
      allow-update { none; };
};

zone "uv.es" IN {
      type master;
      file "master.uv.es";
      allow-update{ none; };
};
```

Y crear a continuación los ficheros "master.147.156", "master.2001:720:1014" y "master.uv.es" con la estructura y los datos de forma similar a la que tienen los ficheros de las zonas obligatorias que vimos con anterioridad.

En concreto, la estructura del fichero "master.147.156" debería ser similar a:

```
$TTL 600            ; 10 minutos
@                   IN SOA glup.irobot.uv.es. root.glup.irobot.uv.es. (
                           2012091300   ; Número de serie.
                           86400        ; Tiempo de validez.
```

```
                        7200        ; Tiempo de reintento.
                        1209600     ; Tiempo de expiracion.
                        7200        ; Validez de las consultas.
                        )
                    NS          glup.irobot.uv.es.
                    NS          amparo.irobot.uv.es.
...
$ORIGIN 222.156.147.in-addr.arpa.
34                  PTR         amparo.irobot.uv.es.
65                  PTR         glup.irobot.uv.es.
...
$ORIGIN 223.156.147.in-addr.arpa.
157                 PTR         mirror.irobot.uv.es.
...
```

Donde la palabra $ORIGIN permite abreviar las entradas posteriores que se realizan en el fichero, de forma que todas aquellas entradas que siguen al símbolo $ORIGIN verán anexadas al final de la dirección IP especificada los datos que siguen en la línea a $ORIGIN, siempre que esto no se impida de forma explícita mediante un carácter “.” al final. De esta forma, la entrada:

```
34                                  PTR         amparo.irobot.uv.es.
```

Al tener delante *$ORIGIN 222.156.147.in-addr.arpa.* es equivalente a haberla escrito de la forma siguiente:

```
34.222.156.147.in-addr.arpa.        PTR         amparo.irobot.uv.es.
```

La validez de la palabra $ORIGIN termina cuando aparece una nueva palabra $ORIGIN, por lo cual la aparición de *$ORIGIN 223.156.147.in-addr.arpa.*, provoca que, por ejemplo, la entrada:

```
157                                 PTR         mirror.irobot.uv.es.
```

Es equivalente a:

```
157.223.156.147.in-addr.arpa.       PTR         mirror.irobot.uv.es.
```

Mientras que la estructura del fichero “ master.2001:720:1014” sería similar a la siguiente:

```
$TTL 600            ; 10 minutos
@                   IN SOA glup.irobot.uv.es. root.glup.irobot.uv.es. (
                        2012091300   ; Número de serie.
                        86400        ; Tiempo de validez.
                        7200         ; Tiempo de reintento.
                        1209600      ; Tiempo de expiracion.
                        7200         ; Validez de las consultas.
                        )
                    NS          glup.irobot.uv.es.
                    NS          amparo.irobot.uv.es.
...
$ORIGIN 2.2.2.0.4.1.0.1.0.2.7.0.1.0.0.2.ip6.arpa.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0  PTR    mirror.ipv6.uv.es.
...
```

Y la estructura del fichero “master.uv.es” debería ser similar a:

```
$TTL 600            ; 10 minutos
```

```
@                   IN SOA glup.irobot.uv.es. root.glup.irobot.uv.es. (
                          2012091300   ; Número de serie.
                          86400        ; Tiempo de validez.
                          7200         ; Tiempo de reintento.
                          1209600      ; Tiempo de expiracion.
                          7200         ; Validez de las consultas.
                          )
                    NS          glup.irobot.uv.es.
                    NS          amparo.irobot.uv.es.
...
$ORIGIN uv.es.
robotica            CNAME       glup.irobot
irtic               CNAME       glup.irobot
autismo             CNAME       glup.irobot
...
amparo              CNAME       amparo.irobot
glup                CNAME       glup.irobot
mirror              CNAME       mirror.irobot
...
$ORIGIN irobot.uv.es.
amparo              A           147.156.222.34
                    MX          10 amparo
                    MX          20 postin.uv.es.
glup                A           147.156.222.65
                    MX          10 glup
                    MX          20 postin.uv.es.
mirror              A           147.156.223.157
                    AAAA        2001:720:1014:222::2
                    MX          10 mirror
                    MX          20 postin.uv.es.
...
$ORIGIN ipv6.uv.es.
mirror              AAAA        2001:720:1014:222::2
...
```

Lo primero que es necesario resaltar en este ejemplo, es que si uno observa existen dos entradas para el ordenador *mirror* que le hacen corresponder la misma dirección IPv6. Esas entradas, si aplicamos el *$ORIGIN* anterior son *mirror.irobot.uv.es* y *mirror.ipv6.uv.es*. Esto permite que ese ordenador responda una dirección IPv6 tanto si realizamos una consulta al dominio *uv.es* (pues *mirror.uv.es* es un alias de *mirror.irobot.uv.es*), como si la consulta es al dominio *ipv6.uv.es*, lo cual posibilita la compatibilidad del nombre *mirror* para los registros de tipo *A* y *AAAA*.

Además, debemos resaltar que si en algunas líneas no aparece el nombre del ordenador se toma, por defecto, el nombre del último ordenador aparecido, por lo que las líneas:

```
amparo                  A           147.156.222.34
                        MX          10 amparo
                        MX          20 postin.uv.es.
```

Se refieren todas ellas al ordenador *amparo*.

Además, podemos observar que las entradas de intercambiador de correo, además de llevar cada línea el nombre de un ordenador (“amparo.irobot.uv.es” y “postin.uv.es” respectivamente), están precedidas de dos valores numéricos. Dichos valores numéricos indican el orden de prioridad de utilización de los intercambiadores de correo, de forma que cuanto menor es el número, mayor es la prioridad de ese intercambiador de correo para el correo destinado a ese ordenador. Resaltar que el valor

numérico introducido delante del intercambiador de correo no importa, solo importa la relación de orden entre los valores.

Por este motivo, un correo destinado a “amparo.irobot.uv.es” se intentará, en primer lugar, entregar al propio “amparo.irobot.uv.es”, y solo en el caso de que dicha entrega no sea posible por cualquier motivo (no tener activado el servicio, etc.), se hará entrega del mismo a “postin.uv.es”.

Servidor secundario de DNS.

Hemos visto como configurar un servidor primario de DNS. Sin embargo, en un dominio suele existir un solo servidor maestro y varios servidores secundarios. La configuración de un servidor secundario es idéntica a la de un servidor primario excepto que las entradas de las zonas de las que se es servidor secundario, por ejemplo del dominio “uv.es”, se introducen como:

```
zone "156.147.in-addr.arpa" IN {
      type slave;
      file "slaves/db.147.156";
      masters {
                147.156.1.1;
              };
};

zone "4.1.0.1.0.2.7.0.1.0.0.2.ip6.arpa" IN {
      type slave;
      file "slaves/db.2001:720:1014";
      masters {
                147.156.1.1;
              };
};

zone "uv.es" IN {
      type slave;
      file "slaves/db.uv.es";
      masters {
                147.156.1.1;
              };
};
```

Donde puede verse que el servidor es un servidor secundario que obtiene las tablas de otro servidor (*type slave*) de ambos dominios. A continuación se le indican los nombres que deben tener los ficheros obtenidos cuando se efectúen las transferencias de zona de ambos ficheros (“slaves/db.147.156”, “slaves/db.2001:720:1014” y “slaves/db.uv.es” respectivamente)[25], y por último aparece la entrada *masters*, dentro de la cual figuran las direcciones IP de los servidores maestros a los que podemos solicitar los ficheros de resolución directa e inversa cuando deseemos solicitar la ejecución de una transferencia de zona (en este caso el ordenador de IP 147.156.1.1).

[25] Estos nombres no tienen porque corresponder con los que tienen los ficheros en el servidor maestro del cual son descargados.

Comentar por último, que las respuestas de un servidor secundario son exactamente igual de validas que las respuestas de un servidor primario, con la única diferencia de que pueden estar sin actualizar con los cambios realizados desde la última transferencia de zona.

Numero de serie en los registros SOA.

Los números de serie, que hemos introducido en los ficheros de los que somos servidores primarios indican la versión del fichero, la cual suele construirse con el formato aaaammdd##, esto es, año, mes, día y dos dígitos, lo que permite un total de 100 modificaciones diarias (dígitos 00 a 99).

El número de serie permite minimizar las transferencias de zona entre los servidores maestros y esclavos, pues posibilita a los servidores esclavos conocer la versión actual que posee el servidor maestro, y actualizar la suya en función de si la versión del servidor maestro es más actual que la que posee el servidor secundario.

Así, si un servidor esclavo posee como número de serie de una de sus zonas 2010110300, y el servidor maestro le indica que su versión es 2010110303, el servidor esclavo solicitará una transferencia de los ficheros de la zona, mientras que si el servidor maestro le responde con el mismo número de serie (o menor, cosa improbable), el servidor esclavo no solicitará la transferencia de los ficheros de zona al estar sus ficheros actualizados.

El programa rndc.

El programa */usr/sbin/rndc* permite enviar ordenes al servidor *named* mientras se ejecuta, si así se ha configurado previamente.

El fichero de configuración por defecto es */etc/rndc.conf*, cuya sintaxis es similar a la de */etc/named.conf* pero mucho más sencilla. En concreto, el fichero */etc/rndc.conf* posee solamente tres secciones:

- La sección *options* permite especificar valores por defecto, y puede tener en su interior tres valores: *default-server*, para especificar el servidor al que se enviaran por defecto los comandos, *default-key*, que permite especificar la clave por defecto que se utilizará para autenticar los comandos, si no se especifica ninguna otra, y *port*, que es el puerto por defecto al que se enviaran los comandos, si no se especifica ningún puerto por defecto se asume el 953 TCP. Solo puede existir una sección *options*.

- La sección *server* <*nombre/ip servidor*> especifica el nombre o dirección IP del servidor al que se aplican las entradas de esta sección. Puede haber una entrada *server* para cada servidor que deseemos especificar. En su interior puede tener en su interior dos valores, *default-key* y *port* con el mismo sentido que las de las sección *options*, pero aplicadas unicamente a este servidor. Pueden existir varias secciones *server*, una por cada servidor que se desee configurar.

- La sección *key* <*nombre*> contiene la identificación de una clave de autenticación. Puede tener en su interior dos valores: *algorithm*, que indica el algoritmo de encriptación utilizado (actualmente solo *hmac-md5* es permitido) y

secret, que contiene, entre comillas, la codificación en base 64 de la clave de encriptación, que puede generarse de forma aleatoria utilizando el comando *rndc-confgen*[26]. Pueden existir varias secciones *key*, una por cada clave de autenticación que se desee especificar.

Un ejemplo de fichero de configuración */etc/rndc.conf* es el siguiente:

```
key "rndc-key" {
     algorithm hmac-md5;
     secret "CLAVE SECRETA CODIFICACADA EN BASE 64";
};

options {
     default-key     "rndc-key"
     default-server  127.0.0.1;
     default-port    953
};
```

Donde la opción *key* especifica la clave de autenticación de nombre *rndckey* y que es la que especifica el servidor local (*127.0.0.1*). Esta clave debe incluirse también en el fichero de configuración */etc/named.conf* para que el servidor de DNS la conozca en el arranque, y pueda identificar al programa *rndc* como autorizado para ejecutar comandos sobre el servidor.

El programa *rndc* posee una serie de opciones que permiten modificar el fichero de configuración por defecto, el servidor por defecto especificado en el citado fichero, etc., por lo que la sintaxis del comando es la siguiente:

```
rndc [opciones] comando
```

Donde las opciones existentes son:

Opción	Descripción	Valor por defecto
-b	Utilizar la dirección indicada para conectarse al servidor.	Dirección IP por defecto.
-c	Utilizar el fichero de configuración indicado en lugar del fichero por defecto.	/etc/rndc.conf
-k	Utilizar el fichero con las llaves indicado en lugar del por defecto.	/etc/rndc.key
-s	Servidor al que se enviará el comando.	El servidor indicado en la opción por defecto de /etc/rndc.conf.
-p	Puerto al que se enviará el comando.	953 TCP
-V	Habilitar modo de depuración.	No habilitado.

[26] Si se desea más información sobre como generar una clave con *rndc-confgen* puede consultarse el manual del sistema.

Opción	Descripción	Valor por defecto
-y	Clave a utilizar de las existentes en el fichero de configuración.	Clave por defecto especificada para el servidor indicado, o si no existe una clave para ese servidor, clave por defecto.

Los comandos que permite ejecutar rndc sobre el servidor named son los siguientes:

Comando	Descripción
`reload`	Releer los ficheros de configuración y las zonas.
`reload <zona>`	Releer la zona especificada.
`refresh <zona>`	Refrescar para mantenimiento la zona especificada.
`freeze`	Suspender la actualización de todas las zonas.
`freeze <zona>`	Suspender la actualización de una zona.
`thaw`	Habilitar la actualización de todas las zonas y releerlas.
`thaw <zona>`	Habilitar la actualización de una zona y releerla.
`reconfig`	Releer el fichero de configuración y las nuevas zonas.
`stats`	Escribir las estadísticas del servidor en el fichero de estadísticas.
`querylog`	Activar el log.
`dumpdb`	Volcar el estado de la cache al fichero named_dump.db.
`stop`	Salvar las actualizaciones pendientes de los ficheros maestros y detener el servidor.
`halt`	Detener el servidor sin salvar las actualizaciones pendientes.
`trace`	Incrementar en una unidad el valor de debug.
`trace <nivel>`	Cambiar el nivel de debug al especificado.
`notrace`	Cambiar el nivel de debug a cero (no debug).
`flush`	Volcar todas las caches de los servidores.
`status`	Mostrar el estado de un servidor.

Un ejemplo de ejecución de un comando es el siguiente:

```
rndc status
```

Obteniendo como respuesta:

```
number of zones: 10
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF: 0
```

```
server is up and running
```

Ejercicios.

1- Configurar los ficheros necesarios para que un ordenador sea cliente del servidor de nombres de los servidores de nombres de la UV (ordenadores 147.156.1.1 y 147.156.1.3), permita añadir de forma automática el dominio uv.es, en las búsquedas de nombres de ordenadores que se realicen y permita resolver las búsquedas del ordenador falsificado.empresilla.com con la dirección IP 147.156.222.65.

2- Configurar un servidor de DNS de forma que sea servidor secundario de los dominios “0.168.192.in-addr.arpa” y “empresilla.com”, cuyo servidor primario tiene la dirección IP 192.168.0.1.

3- Configurar un servidor de DNS de forma que actúe como servidor primario de los dominios “0.168.192.in-addr.arpa” y “empresilla.com”, y como servidor secundario del dominio uv.es, cuyo servidor primario es 147.156.1.1.

4- Una empresa desea registrar en Internet el dominio “empresilla.com”, cuyas direcciones IP corresponden al rango 192.168.0.0/24, contando inicialmente con los siguientes ordenadores:

- Un servidor web, de nombre www.empresilla.com, que también actúa como servidor de ftp, por lo que también recibe el nombre de ftp.empresilla.com. Además, este servidor es el servidor secundario de correo, por lo que recibe el correo si el servidor de correo se encuentra inoperativo.

- Un servidor de correo, de nombre correo.empresilla.com, que recibe todo el correo de la empresa. Además este servidor actúa como servidor secundario de DNS.

- Un servidor primario de DNS, de nombre dns.empresilla.com.

- Cinco PCs de nombres pc1, …, pc5.

Configurar todos los servidores de DNS y ficheros necesarios para poder registrar el dominio en Internet. Como ayuda se os recuerda que para poder registrar el dominio en Internet deben existir, como mínimo, dos servidores de DNS en el dominio.