



2023

2024



Módulo: Despliegue de Aplicaciones Web



Unidad de Trabajo: 1

Servicios de red implicados en el despliegue de una aplicación



Contenido de la unidad

1	Introducción	3
2	Sistema de nombre de dominio	3
2.1	Resolución	4
2.2	Nombre de dominio	4
2.3	Niveles de dominio	4
3	Zonas de búsquedas, tipos de servidores DNS y registros	6
3.1	Tipos de Servidores DNS	6
3.2	Registros DNS	7
4	Instalación y configuración de un servicio DNS (Linux)	7
4.1	Configuración de un DNS Maestro	8
4.1.1	Configurar el archivo de opciones (named.conf.options)	8
4.1.2	Configurar el archivo local	9
4.1.3	Crear el archivo de la zona directa	9
4.1.4	Crear los archivos de la zona inversa	10
4.2	Configuración de un DNS Esclavo	11
4.3	Utilización de reenviadores externos	11
5	Comprobación del DNS	12
5.1	nslookup	12
5.2	dig	12
6	Otros comandos de interés	13
7	Anexo - Base de Datos	14
7.1	Estructura	14
7.1.1	Tipo de registro SOA.	15
7.1.2	Tipo de registro NS	15
7.1.3	Tipo de registro A, PTR, CNAME Y MX.	16
8	Actividades	17

1 Introducción

Una aplicación web necesita de servicios de red para poder funcionar de forma correcta y coherente. Estos servicios son el **servicio DNS** y el **servicio de directorio LDAP**.

Uno de los servicios más críticos es el servicio DNS, ya que es necesario en cualquier dispositivo que desee conectarse a Internet y acceder a los recursos mediante en nombre de dominio, en lugar de la dirección IP.

El servicio DNS está dividido en dos zonas:

- **Zona directa**
- **Zona inversa**

En cada zona se definen distintos tipos de registros para que el servicio DNS permita resolver las peticiones de los clientes. Por otro lado, también existen distintos tipos de servidores DNS en función de nuestras necesidades.

En cuanto al servicio de directorio LDAP, podemos decir que es un software que permite la validación de usuarios de forma centralizada y permite controlar el acceso a cualquier aplicación que esté instalada en el sistema operativo. Tiene una estructura jerárquica y centralizada

2 Sistema de nombre de dominio

En las **comunicaciones en red TCP/IP**, cuando un equipo envía un paquete de datos a otro, **es necesario identificar tanto el origen como el destino**. Las aplicaciones que vayan a iniciar esta transmisión deben incorporar valores obligatorios tales como: dirección MAC de origen y de destino, dirección IP de origen y de destino y puerto de origen y de destino.

Estos campos siempre deben tener un valor asignado. Significa que, cuando se quiere visitar una página web que está alojada en un equipo (por ejemplo, el equipo con el nombre `www.juntadeandalucia.es`), el cliente debe saber cuál es la dirección IP de destino para incorporarla en el paquete.

El **servicio de nombres de dominio (DNS)** es el encargado de averiguar qué dirección IP correspondería a cada nombre. De esta manera, cuando un equipo solicite un recurso especificando su nombre, el DNS devuelve su dirección IP, o viceversa. La ventaja principal que tiene el servicio DNS es la simplificación y la comodidad. Obviamente, es muy difícil para el ser humano recordar todas las direcciones IP de los equipos que ofrecen algún servicio, y más cuando estas direcciones cambian cada cierto tiempo. El poder acceder por nombre es mucho más fácil de recordar que utilizando números.

Realmente, cuando se abre un navegador web y se especifica una URL, se puede disfrutar de un servicio web tanto por nombre como por dirección IP.

Para ello, basta con ejecutar el **comando ping** en un servidor conocido, por ejemplo, `www.google.com`. En la primera línea de la ilustración se puede observar la dirección IP de respuesta.

```
C:\Users\Miguel Ángel>ping www.google.com  
  
Haciendo ping a www.google.com [172.217.17.4] con 32 bytes de datos:  
Respuesta desde 172.217.17.4: bytes=32 tiempo=15ms TTL=114  
Respuesta desde 172.217.17.4: bytes=32 tiempo=17ms TTL=114
```

Ilustración 1 - Dirección IP de respuesta tras ejecutar el comando ping en `www.google.com`

Si se coloca esa misma dirección IP en un navegador web, se puede comprobar que devuelve la página web de inicio.

2.1 Resolución

Casi la mayoría de la actividad de Internet se basa en los DNS, que permiten recuperar velozmente la información correspondiente para poderte conectar al servidor o host remoto.

Existen básicamente dos formas para que el dispositivo resuelva el nombre de dominio requerido, y son las siguientes:

- 1) En el **fichero /etc/hosts**¹ de cualquier SO Linux
- 2) La segunda forma es la relacionada **con los servidores DNS**, que hay que escribir en la configuración de la tarjeta de red manualmente. Si existiera el servicio de DHCP, no sería necesario escribirlos manualmente.

De estas dos formas, la más habitual es la segunda, ya que no es operativo incluir en el fichero de host todas las direcciones IP y nombres a los que accedemos habitualmente.

El funcionamiento de cualquier dispositivo que se conecte a Internet es leer el fichero host y, si no se encuentra la dirección en este fichero, automáticamente se busca en los servidores DNS de forma jerárquica.

2.2 Nombre de dominio

Un **nombre de dominio** (a menudo denominado simplemente dominio), es un nombre fácil de recordar asociado a una dirección IP física de Internet, es decir, una cadena de caracteres que se utiliza para acceder a sitios web y otros recursos en Internet de manera más intuitiva que tener que recordar direcciones IP numéricas.

Un nombre de dominio consta de dos partes principales:

- **Nombre de Host:** Esta es la parte del nombre de dominio que representa un recurso específico, como un sitio web o un servidor de correo electrónico. Por ejemplo, en "www.ejemplo.com", "www" es el nombre de host.
- **Dominio de Nivel Superior (TLD):** El TLD es la parte del nombre de dominio que se encuentra después del nombre de host y generalmente se refiere a la categoría o la ubicación geográfica del sitio web. Algunos ejemplos comunes de TLD son ".com", ".org", ".net", ".gov", ".edu", y ".es" (para España).

Se trata del nombre único que se muestra después del signo @ en las direcciones de correo y después de www. en las direcciones web.

Cualquiera puede comprar un nombre de dominio. Sólo tienes que ir a un registrador o un host de dominios, encontrar un nombre que nadie más utilice y abonar una pequeña cuota anual para ser su propietario.

Un subdominio forma parte de un dominio más grande. Por ejemplo, mail.google.com, www.google.com y docs.google.com son todos los subdominios del dominio google.com. El propietario de un dominio puede crear subdominios en su dominio de nivel superior para que cada uno de sus servicios o páginas web tenga una dirección fácil de recordar.

2.3 Niveles de dominio

La **jerarquía de nombres** está organizada en forma de árbol, de manera que **todos los nombres parten de un nodo raíz** y llegan a un nodo hoja. A la hora de nombrar un equipo, se va navegando por la estructura recorriendo distintos niveles, partiendo de un nivel inferior (más específico) y subiendo por los nodos hasta llegar al nivel más alto (la raíz). **Cada nivel es administrado por una entidad** que establece una normativa que regula el uso de los nombres.

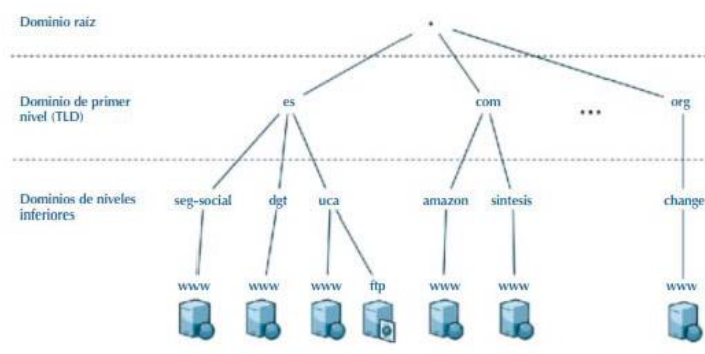


Ilustración 2 - Estructura jerárquica de nombres de dominio

¹ En versiones Windows: C:\Windows\System32\drivers\etc\hosts

Existen varios niveles de dominios en Internet:

- a) **Dominio raíz: representado por un punto (.),** es de donde parten todos los nombres de dominio. **La autoridad superior que se encarga de los nombres y las direcciones IP en internet es la IANA** (Internet Assigned Numbers Authority), actualmente integrada dentro de la **ICANN**. Es la responsable de coordinar y mantener todo el funcionamiento de los DNS a nivel mundial.
- b) **Dominios de nivel superior (TLD):** son dominios que en la estructura jerárquica de nombres DNS se encuentran bajo el dominio raíz. Dentro de este tipo de dominio podemos realizar la siguiente subclasificación:
 - **Dominios genéricos o gTLD:** (sigla en inglés de Generic Top Level Domain) son aquellos que tienen tres o más letras: .com, .org, .info, .pro son algunos ejemplos. Inicialmente pensados para una clase particular de organizaciones (por ejemplo, .com para organizaciones comerciales), actualmente la mayoría de ellos pueden usarse sin restricción. No obstante, se mantienen una serie de ellos para usarse de manera restringida.
 - **Dominios geográficos o ccTLD** (sigla en inglés de Country-Code Top_Level Domain) son los formados por dos letras y en general hacen referencia a un país (Asignados por la ISO 3166-1): .es, .uk, .us, .fr son algunos ejemplos.
 - **Dominio .arpa:** es una excepción y por eso aparece aparte. in-addr.arpa es usado por los servidores DNS inversos para la obtención del FQDN de una dirección IP (búsqueda DNS inversa). Por ejemplo la dirección 212.30.222.56 es mapeada al nombre 56.222.30.212.in-addr.arpa.
- c) **Dominios de segundo nivel:** son los dominios que se encuentran bajo los TLDs. Cada uno de estos dominios está registrado a favor de una determinada entidad (empresa, universidad, órgano, persona, etc.). La entidad propietaria del dominio es la encargada de la gestión del dominio. Para un dominio de este tipo se tienen uno o varios servidores DNS que tienen información sobre máquinas disponibles en el dominio, sobre posibles subdominios y sobre servidores DNS del dominio y de los subdominios. Cuando una entidad desea disponer de un dominio, debe registrarlo ante un registrador oficial autorizado por ICANN.
Dominios de segundo nivel son wikipedia.org, mec.es, google.com y otros muchos.
- d) **Subdominios:** Son dominios que hay bajo un dominio de segundo nivel o bajo otro subdominio. Un subdominio no tiene que ser registrado como un dominio de segundo nivel. Es el propietario del dominio de segundo nivel quien decide la existencia, o no, de subdominios. En un subdominio puede haber servidores encargados de toda la gestión del subdominio aunque también esa gestión se puede llevar a cabo desde los servidores de segundo nivel. Para cada subdominio se tiene información sobre las máquinas y servidores pertenecientes al subdominio.

Para ejemplificar lo anterior, podemos usar el nombre de dominio: `www.kali.org`, en donde:

- **www:** nombre del host / subdominio. Identifica el servidor web que almacena la página web.
- **kali:** es un dominio de segundo nivel, cuyo dominio padre en nuestro caso es org. Identifica de forma coherente al nombre de nuestra organización, empresa, organismo, etc.
- **org:** es el dominio de primer nivel (TLD) que identifica a organizaciones



Ilustración 3 - Diferencia entre dominio y URL

3 Zonas de búsquedas, tipos de servidores DNS y registros

El servicio de DNS, permite una ventaja fundamental de la que carece cualquier opción manual en un fichero, y es que cualquier cambio en la dirección IP, o un nombre, en cualquier dispositivo que esté en Internet, o en una red, se puede replicar a todos los servidores DNS. Por lo que esta funcionalidad y simplicidad hacen que el servicio DNS sea fundamental en cualquier empresa u organización para su funcionamiento a nivel informático.

Además, los cambios son dinámicos, como veremos en un apartado posterior.

Normalmente, los servidores **DNS se componen de zonas** que son las encargadas de contener los diferentes tipos de registros, en función de qué tipo de servicio ofrezca una dirección IP determinada. Por ejemplo, no es lo mismo un servidor de correo, que un servidor web.

Existen dos **tipos de zonas**:

- **Zona de búsqueda directa:** esta zona permite traducir el nombre de dominio a la dirección IP del recurso solicitado. Es la que suele utilizarse por defecto en la resolución de nombres
- **Zona de búsqueda inversa:** los registros que se definen en esta zona permiten obtener un nombre de un dominio a partir de una dirección IP. Principalmente, se utiliza por aplicaciones para temas de seguridad, como, por ejemplo, detección de spam en correo electrónico o comprobación de emisores fraudulentos.

Nota:

- La resolución directa es necesaria para todo funcionamiento en un servicio DNS. La inversa es opcional, pero recomendable.

- Se denomina **zona** al conjunto de nombres gestionados por un dominio.



Las zonas, además de ser de búsqueda directa e inversa, pueden ser de varios tipos, entre los más habituales están:

- **Master:** En una zona *master* el fichero de zona con los registros de recursos (RR Resource Record) se encuentra en el disco local del servidor y sus respuestas sobre el dominio serán autoritarias.
- **Slave:** una zona slave es una réplica de una zona master, donde el fichero de zona se obtiene mediante una operación de transferencia de zona. Las respuestas del servidor sobre la zona serán autoritarias.

3.1 Tipos de Servidores DNS

El **servicio DNS** es un software que permite responder a las peticiones que realizan los clientes y que están estrechamente relacionadas con el espacio de nombres de dominio. Se suele realizar una clasificación como la siguiente:

- **Servidores primarios o maestros:** son servidores que guardan la información relacionada con las zonas de las que son autoritarios. Sus archivos son de lectura y escritura. El administrador es el encargado de añadir, modificar o eliminar los nombres de dominio. Cualquier cambio debería ser notificado a este servidor para que de esta forma tenga la información actualizada.
- **Servidores secundarios o esclavos:** son un tipo de servidor que no tiene los propios archivos de zona, sino que están transferidos de una zona primaria o maestra. Estos servidores actúan cuando el servidor primario o maestro no puede resolver la petición por cualquier causa (servidor no disponible, caído, sin conexión, etc.).
- **Servidores caché:** estos servidores se configuran para mejorar los tiempos de respuesta de las consultas, reducir la carga de los equipos y disminuir el tráfico de la red. Por lo tanto, su función es contactar con otros servidores para resolver las peticiones de los clientes.

Normalmente colocado entre los PC y los servidores de nombres autoritarios, recibe los RR de respuesta de las consultas DNS de los PC y los guarda en su caché durante un tiempo (TTL) para responder más rápido a las mismas preguntas si surgen de nuevo. Cuando la respuesta a la consulta del PC no está en la caché del servidor DNS, este traslada la consulta a un servidor autoritario, y la respuesta la envía al PC como autoritaria, además de guardarla en la caché. Si la misma pregunta llega antes de que transcurra el TTL de los RR de la respuesta, esta se envía directamente al PC, con el consiguiente ahorro de tiempo, pero como no autoritaria, por encontrarse en la caché.

IMPORTANTE: BIND por defecto funciona como servidor DNS caché

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

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

3.2 Registros DNS

Las zonas, con los espacios de nombres, se almacenan en una base de datos formada por uno o más ficheros denominados **registro de recursos** y que pueden estar alojados en uno o más servidores DNS.

Estos ficheros contienen cadenas de texto que definen algún valor dentro de la zona. Cada cadena puede ser de varios tipos:

- **Comentarios:** comienzan con **punto y coma (;)** e incluyen notas aclaratorias sobre el fichero.
- **Directivas:** valores que especifican ciertos aspectos del RR. Comienzan con el **símbolo del dólar (\$)**. Las directivas más utilizadas son:
 - **\$ORIGIN:** define el nombre de dominio que será incluido al final de cualquier nombre que se defina en el RR y que no acabe en el punto (.). Esta directiva no es obligatoria y, si se omite, se tomará como valor la cadena especificada en la sección zone del fichero /etc/bind/named.conf, que se explicará más adelante.
 - **\$TTL:** es el valor del Time to Live predeterminado para una zona. Está expresado en segundos e indica el tiempo que un RR es válido. Cada registro de recurso puede tener su propio TTL, ignorándose esta directiva.
- **Registro de recurso:** líneas de texto que sirven para definir diversas entidades dentro del dominio. Las más utilizadas son:

TIPO	Descripción	Sintaxis
SOA	Registro que define el comienzo de la zona. Debe estar colocado directamente detrás de las directivas y define información importante acerca de la autoridad de los RR para la zona.	@ IN SOA <servidor de nombres primario> <email del administrador> (<número de serie> <tiempo de refresco> <tiempo de reintento> <tiempo de expiración> TTL mínimo)
NS	Registro que define los nombres de los servidores autoritarios para una zona en particular.	IN NS <nombre del servidor>
A	Define y asigna una dirección IPv4 a un nombre.	<host> IN A <dirección IP>
AAAA	Define y asigna una dirección IPv6 a un nombre. <host>	<host> IN AAAA <dirección IP>
MX	Define un servidor de correo electrónico para el dominio	IN MX <Valor preferencia> <nombre servidor email>
CNAME	Nombre canónico, define alias a nombres de dominio, enlazando nombres. Estos nombres de dominio deben tener un registro A.	<nombre alias> IN CNAME <nombre real>
PTR	Traduce direcciones IP en nombre de dominio	<IP-host> IN PTR <host>

4 Instalación y configuración de un servicio DNS (Linux)

La instalación que se va a realizar es del **servicio bind9 y sus utilidades**. El procedimiento de instalación es el siguiente:

- 1) Antes de comenzar a instalar el paquete bind9 (servicio DNS), se debería configurar una IP en la máquina o PC en que se va a instalar este servicio, y tener acceso a Internet. (**En caso de dudas, ver tema 0**)
- 2) Una vez tenemos conexión a Internet, ejecutaremos los comandos:
 - **# apt-get update:** actualiza la lista de paquetes disponibles y versiones, pero no actualiza ningún paquete. Esta lista se selecciona de los servidores con repositorios que se definen en el sources.list.
 - **# apt-get upgrade:** instalará las nuevas versiones respetando la configuración del software, cuando sea posible.
- 3) Una vez actualizado el repositorio y nuestro sistema, podemos proceder a instalar el bind9 y sus utilidades con el siguiente comando:


```
# apt-get install bind9 bind9utils
```


- 4) Una vez instalado, comprobaremos que el servicio bind9 se instaló correctamente en el sistema. Para ello se usan los siguientes comandos.


```
# service bind9 start
# service bind9 status
```
- 5) Una vez comprobado que el servicio está correcto, se está en disposición de configurar nuestro servidor DNS, no sin antes realizar una copia de seguridad de los ficheros de configuración que se van a modificar, por si es necesario deshacer los cambios realizados y comenzar desde el principio.
 El primer fichero que lee el servicio bind9 es el **/etc/bind/named.conf**. En un principio, no es necesario modificar nada en él. Este fichero incluye tres ficheros importantes que va a leer el servicio bind9, que son los siguientes:
 - **/etc/bind/named.conf.options**: en este primer fichero se define la caché de nuestro DNS, y la configuración genérica del servidor, como puede ser la transferencia de zonas, forwarders, etc. La configuración más importante de este fichero son los forwarders (reenviadores), ya que es necesario configurarlos para que cuando un cliente realice una consulta DNS y no encuentre la respuesta en local, la respuesta se pueda encontrar en Internet, como puede ser el nombre de un servidor, un sitio web, un servidor de correo, etc. Por lo general, son los servidores DNS de nuestro ISP, Google, Telefónica, etc.
 - **/etc/bind/named.conf.local**: este fichero es tan importante como el anterior, y es donde se definen las zonas de búsqueda directa e inversa que va a gestionar nuestro servicio DNS.
 - **/etc/bind/named.conf.default-zones**: es un fichero donde se definen las zonas por defecto. La inclusión de este fichero se puede comentar. Realmente donde se definen las zonas nuevas es en el fichero **named.conf.local**.

4.1 Configuración de un DNS Maestro

La configuración de BIND consta de varios archivos que se incluyen / se cargan desde el archivo de configuración principal, **named.conf**. Estos nombres de archivos comienzan con named porque ese es el nombre del proceso que BIND ejecuta (abreviatura de “domain name daemon”).

Punto de partida de un caso práctico

Para comprender mejor la configuración del DNS, vamos a suponer que queremos configurar nuestro DNS para que gestione el dominio “**ejemplo.com**”.

Para ello vamos a tener tres servidores 192.168.0.1/24 (ns1 - DNS Maestro), 192.168.0.2/24 (ns2 - DNS Esclavo) y 192.168.0.3/24 (ns3 - DNS Esclavo).

4.1.1 Configurar el archivo de opciones (named.conf.options)

En la máquina donde está instalado el bind (por ejemplo, server1), accedemos al fichero **/etc/bind/named.conf.options** para su edición.

Sobre el bloque options existente, crearemos un nuevo bloque ACL (lista de control de acceso) llamado por ejemplo “dns-esclavos”. Aquí definiremos una lista de clientes desde los que permitiremos consultas de DNS recurrentes (es decir, nuestros servidores DNS, o los clientes desde los que autorizamos a hacer consultas directas al DNS).

Usando nuestras direcciones IP privadas de ejemplo, añadiremos server2 y server3 a nuestra lista de clientes de confianza:

```
acl "dns-esclavos" {
    192.168.0.2;    # ns2 - (server2)
    192.168.0.3;    # ns3 - (server3)
    192.168.0.0/24; # Si queremos añadir toda la red
};
```

```
options {
...
```

Ahora que tenemos nuestra lista de clientes DNS de confianza, nos convendrá editar el bloque options. En este momento, el inicio del bloque tiene el siguiente aspecto:

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


Debajo de la directiva `directory`, añadiremos las siguientes líneas

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

    version "No disponible"           # Por seguridad, evitaremos dar pistas de la versión de BIND
    recursion yes;                    # Permitirá consultas recursivas
    allow-recursion { dns-esclavos; }; # Permite consultas recursivas a los clientes
                                      # de la lista " dns-esclavos "

    listen-on port 53{ 192.168.0.1; }; # Puerto e IP del servidor DNS por el que se escucha
    listen-on port 53{ any; };        # El DNS escucha por todas sus interfaces y por el puerto 53

    allow-transfer { none; };         # Por defecto vamos a desactivas la transferencia de zonas.
                                      # Será desde cada zona, desde donde se permitirán a los
                                      # equipos que interese.

    forward only;
    forwarders {                      # Se utiliza para definir la lista de direcciones IP a las
        8.8.8.8;                      # que reenviar las consultas.
        8.8.4.4;                      # Solo es tenido en cuenta si se usa el estamento forward
    };
    ...
};
```

Una vez finalizado, guardamos los cambios

4.1.2 Configurar el archivo local

Cada vez que queramos definir una nueva zona primaria, tendremos que editar el fichero `/etc/bind/named.conf.local`. En dicho fichero definiremos las distintas zonas (directa e inversa) e indicaremos el fichero que tiene la información para la resolución de nombres de dominio.

NOTA: Es recomendable hacer una copia de seguridad del fichero antes de modificarlo.

El formato para definir cada una de las zonas que nuestro DNS gestionará será:

```
zone "dominio" {
    type master;
    notify yes | no;
    allow-transfer { IP_1; IP_2; ...; IP_N; | ACL; | none; }; #IP DNS secundarios
    file "/etc/bind/fichero_de_base_de_datos";
};

zone "dominio.in-addr.arpa" {
    type master;
    notify yes | no;
    allow-transfer { IP_1; IP_2; ...; IP_N; | ACL; | none; }; #IP DNS secundarios
    file "/etc/bind/zones/fichero_de_base_de_datos";
};
```

Si configuras una zona Primaria para el dominio “ejemplo.com” dentro de nuestra red 192.168.0.0/24, tendremos que agregar:

```
zone "ejemplo.com" {
    type master;
    notify no;
    allow-transfer { none; };
    file "/etc/bind/db.ejemplo.com";
};




zone "0.168.192.in-addr.arpa" {
    type master;
    notify no;
    allow-transfer { none; };
    file "/etc/bind/zones/db.192.168.0";
};
```

4.1.3 Crear el archivo de la zona directa

Para crear el fichero de resolución directa para nuestra zona, vamos a utilizar un archivo de zona existente como plantilla. Siguiendo con nuestro ejemplo, tendremos que crear el archivo `/etc/bind/db.ejemplo.com` (tal y como se ha indicado en el fichero de definición de zonas que va a gestionar nuestro DNS)

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

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

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

También, crea el registro NS y el registro A para ns1.ejemplo.com. (El servidor de nombres en este ejemplo).

```
;
; BIND data file for ejemplo.com
;
$TTL 604800
$ORIGIN ejemplo.com.

@ IN SOA ns1.ejemplo.com. root.ejemplo.com. (
    2020110200 ; Serial
    604800     ; Refresh
    86400      ; Retry
    2419200    ; Expire
    604800 ) ; TTL
;
; Registros NS
@ IN NS ns1.ejemplo.com. ; Servidor DNS Primario
@ IN NS ns2.ejemplo.com. ; Servidor DNS Secundario
@ IN NS ns3.ejemplo.com. ; Servidor DNS Secundario

; Registros A
ns1.ejemplo.com. IN A 192.168.0.1
ns2 IN A 192.168.0.2
ns3 IN A 192.168.0.3
www IN A 192.168.0.10
ftp IN CNAME www
```

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

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

Nota: A muchos administradores les gusta utilizar la última fecha de edición como la serie de una zona, así como 2020110200 que es yyymmddss (donde ss es el Número de Serie).

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

```
# sudo service bind9 restart
```

4.1.4 Crear los archivos de la zona inversa

Al igual que se hizo para crear el archivo de zona directa, vamos a usar /etc/bind/db.127 como plantilla para crear el fichero de resolución inversa.

```
sudo cp /etc/bind/db.127 /etc/bind/db.192.168.0
```

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

```
;
; BIND reverse data file 0.168.192.in-addr.arpa.
;
$TTL 604800
$ORIGIN 0.168.192.in-addr.arpa.

@ IN SOA ns1.ejemplo.com. root.ejemplo.com. (
    2020110200 ; Serial
    604800     ; Refresh
    86400      ; Retry
    2419200    ; Expire
    604800 ) ; TTL
;
;
; Registros NS
@ IN NS ns1.ejemplo.com. ; Servidor DNS Primario
@ IN NS ns2.ejemplo.com. ; Servidor DNS Secundario
@ IN NS ns3.ejemplo.com. ; Servidor DNS Secundario

; Registros PTR
1. 0.168.192.in-addr.arpa. IN PTR ns1.ejemplo.com.
2 IN PTR ns2.ejemplo.com.
3 IN PTR ns3.ejemplo.com.
10 IN PTR www.ejemplo.com.
10 IN PTR ftp.ejemplo.com.
```

El número de serie en la zona inversa debe incrementarse con cada cambio. Para cada entrada "A" que configuras en "/etc/bind/db.ejemplo.com" debes crear una entrada "PTR" en "/etc/bind/db.192.168.0".

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

```
# sudo service bind9 restart
```

4.2 Configuración de un DNS Esclavo

Una vez que has creado una zona primaria, es necesaria una zona secundaria para poder mantener la disponibilidad del dominio si el primario se vuelve no disponible. Recuerda que una zona secundaria contiene una copia de una zona primaria. Un servidor secundario actualiza la zona cada cierto tiempo por transferencia de zona desde el servidor primario de la zona. Primeramente, en el servidor maestro primario, se necesita permitir la transferencia de la zona. Añade la opción **allow-transfer** y **notify** a las definiciones de ejemplo de zona directa e inversa en `/etc/bind/named.conf.local`:

```
zone "ejemplo.com" {
    type master;
    notify yes;
    allow-transfer { 192.168.0.2; 192.168.0.3; };      # allow-transfer { dns-esclavos; };
                                                    # Si hemos definido ACL
    file "/etc/bind/db.ejemplo.com";
};

zone "0.168.192.in-addr.arpa" {
    type master;
    notify yes;
    allow-transfer { 192.168.0.2; 192.168.0.3; };      # allow-transfer { dns-esclavos; };
                                                    # Si hemos definido ACL
    file "/etc/bind/db.192.168.0";
};
```

A continuación, en el DNS secundario, instala el paquete `bind9` de la misma forma que para el primario. Luego, edita `/etc/bind/named.conf.local` y añade las siguientes declaraciones para la zona directa (Forward) e inversa (Reverse). Observe que el tipo es "slave", el archivo no contiene una ruta y hay una directiva `masters` en la que debería fijarse la dirección IP del servidor DNS primario.

```
zone "ejemplo.com" {
    type slave;
    masters { 192.168.0.1; };                        # Indicamos la IP del servidor DNS primario
    allow-notify { 192.168.0.1; };                    # Por seguridad, solo aceptaremos mensajes NOTIFY del maestro
    file "db.ejemplo.com";                            # El fichero de zona será /var/cache/bind/db.ejemplo.com
};

zone "1.168.192.in-addr.arpa" {
    type slave;
    masters { 192.168.0.1; };                        # Indicamos la IP del servidor DNS primario
    allow-notify { 192.168.0.1; };                    # Por seguridad, solo aceptaremos mensajes NOTIFY del maestro
    file "db.192.168.0";                            # El fichero de zona será /var/cache/bind/db.192.168.0
};
```

Nota: También tendremos que hacer los cambios en el fichero `named.conf.options`, tal y como se hizo en el DNS primario, pero en el `listen-on` deberás poner la IP del servidor secundario que estás configurando

Reinicia BIND9 en el DNS secundario:

```
# sudo service bind9 restart
```

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

4.3 Utilización de reenviadores externos

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

Si queremos que nuestro servidor DNS utilice reenviadores externos como los que tienen las direcciones IP 195.235.1133 y 195.235.96.90 simplemente tenemos que editar el archivo de configuración `/etc/bind/named.conf.options` y escribir en la directiva `forwarders` que hay dentro de `options {}`, las direcciones de los reenviadores.

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

    version "No disponible"                          # Por seguridad, evitaremos dar pistas de la versión de BIND
    recursion yes;                                    # Permitirá consultas recursivas
    ...
    forward only;
    forwarders {                                     # Se utiliza para definir la lista de direcciones IP a las
        8.8.8.8;                                     # que reenviar las consultas.
        8.8.4.4;                                     # Solo es tenido en cuenta si se usa el estamento forward
    };
    ...
};
```

5 Comprobación del DNS

A continuación, tendremos que comprobar que el DNS está funcionando correctamente con los registros que hemos declarado. Las pruebas se pueden hacer con varios comandos de cliente DNS, como pueden ser **nslookup**, **dig** o **host**. En nuestro caso, vamos a usar dos comandos **nslookup** y **dig** para que se observen las distintas opciones, una más breve y otra más extensa.

5.1 nslookup

El **nslookup** es un programa para consultar servidores DNS. Se utiliza para saber si un servidor DNS resuelve correctamente los nombres DNS y las direcciones IP, para solucionar problemas frecuentes de los servidores DNS o, para diagnosticar problemas ocasionales de configuración en los servidores DNS.

Con nslookup podemos obtener la dirección IP asociada a un nombre DNS y viceversa, además, podemos preguntar a los servidores de nombres información relativa a los registros de recursos (RR) de la/s zona/s de las que son autorizados.

Nota: Este comando funciona tanto en sistemas operativos UNIX/Linux como en Windows.

Hay dos formas de ejecutar nslookup:

- ✚ **Modo normal o no interactivo:** se usa para presentar sólo el nombre y la información solicitada para un host o nombre DNS.
- ✚ **Modo interactivo:** permite al usuario consultar los servidores DNS para obtener información sobre varios hosts y dominios o para listar los hosts de un dominio. Esta opción la dejaremos para que investigue el alumno, ya que se sale de los objetivos a alcanzar en este módulo.

Para nuestro estudio, la sintaxis que vamos a usar de este comando sería:

```
$ nslookup <name> [server]
```

- El primer argumento "name", es el nombre de host o la direcciones IP del host a buscar.
- El segundo argumento (server), es opcional y especifica el nombre de host o la dirección IP del servidor DNS al que se va a consultar. En caso de no indicarlo, será el DNS que tenga por defecto la máquina desde la que hacemos la consulta.

Ejemplo1: Consultar por nombre DNS al servidor DNS configurado por defecto en las propiedades TCP/IP del equipo:

```
$ nslookup ns1.ejemplo.com
```

Ejemplo2: Consultar por una dirección IP al servidor DNS configurado por defecto en las propiedades TCP/IP del equipo:

```
$ nslookup 192.168.0.1
```

Ejemplo2: Preguntar al servidor DNS 8.8.8.8 por el nombre de dominio www.google.es

```
$ nslookup www.google.es 8.8.8.8
```

Nota:

- **Authoritative Answer:** significa que la respuesta DNS se ha producido desde el servidor DNS que tiene todo el archivo de información disponible para esa zona.
- **Non Authoritative Answer:** significa que la respuesta DNS se ha producido desde un servidor DNS que tiene en caché una copia de las consultas realizadas para esa zona, al servidor que tiene la Autoridad para responder (el que tiene el archivo de zona). Por esto veremos muy a menudo la respuesta desde servidores que son Non Authoritative.

5.2 dig

El **comando dig** (Domain Information Groper) es un programa utilizado para preguntar a los servidores DNS.

Normalmente, dig se usa pasándole argumentos desde la línea de comandos (CLI), pero también tiene un modo de operar por lotes, leyendo las consultas desde un archivo.

Con la opción -h presenta un resumen de sus argumentos y opciones para usar desde la línea de comandos (CLI).

A menos que se especifique un servidor DNS concreto, dig preguntará a cada uno de los servidores DNS que tenga configurado el cliente.

Si no se pasan argumentos u opciones en la línea de comandos (CLI), dig realizará una consulta de los registros de recursos (RR) tipo NS para el dominio raíz (.).

Para nuestro estudio, la sintaxis que vamos a usar de este comando sería:

```
$dig [-h] [@server] [name] [-x addr]
```

- **server:** es el nombre o la dirección IP del servidor DNS a consultar. Puede ser una dirección IPv4 en notación decimal con puntos o una dirección IPv6 en notación delimitada por dos puntos. Cuando el argumento del servidor proporcionado es un nombre de host (hostname), dig resuelve ese nombre antes de consultar ese servidor DNS. Si no se proporciona ningún argumento de servidor, dig usará los DNS que tenga configurados el ordenador desde el que se está haciendo la consulta. Se muestra la respuesta del servidor DNS que responde.
- **name:** es el nombre DNS que deseamos buscar.
- **addr:** La IP, de la que deseamos conocer su nombre de dominio.

Ejemplo1: Consultar por nombre DNS al servidor DNS configurado por defecto en las propiedades TCP/IP del equipo:

```
$ dig ns1.ejemplo.com
```

Ejemplo2: Consultar por una dirección IP al servidor DNS configurado por defecto en las propiedades TCP/IP del equipo:

```
$ dig -x 192.168.0.1
```

Ejemplo2: Preguntar al servidor DNS 8.8.8.8 por el nombre de dominio www.google.es

```
$ dig @8.8.8.8 www.google.es
```

6 Otros comandos de interés

ipconfig (Widnows)

En relación con el servicio DNS, este comando permite:

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

systemd-resolve

En las últimas distribuciones de Ubuntu / Debian, se emplea el comando `systemd-resolved` para la resolución de DNS.

En relación con el servicio DNS, este comando permite:

- Mostrar las estadísticas de la caché del DNS: `systemd-resolve --statistics`
- Vaciar la caché de DNS: `systemd-resolve --flush-caches`
- Mostrar información de los servidores DNS: `systemd-resolve --status`

named-checkconf

El comando `named-checkconf` sirve para chequear la sintaxis de los ficheros de configuración de BIND. En el chequeo incluye aquellos ficheros de la instrucción `include`. Su sintaxis es la siguiente:

```
# named-checkconf [ fichero ]
```

Si no se especifica ningún fichero, chequeará el fichero `named.conf` junto con todos los ficheros `include` que tenga.

```
# named-checkconf /etc/bind/named.conf.options
```

```
# named-checkconf
```

named-checkzone

El comando `named-checkzone` se utiliza para chequear la sintaxis de un fichero de zona. Su sintaxis es la siguiente:

```
# named-checkzone <nombre-zona> <fichero>
```

Por ejemplo:

```
# named-checkzone ejemplo.com /etc/bind/db.ejemplo.com
```

7 Anexo - Base de Datos

7.1 Estructura

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

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

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

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

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

Donde:

- ✚ **Propietario:** es el nombre de host o del dominio DNS al que pertenece este recurso. Puede contener:
 - Un nombre de host/dominio completamente cualificado o no.
 - El símbolo "@" que representa el nombre de la zona que se está describiendo.
 - Una cadena vacía, en cuyo caso, equivale al propietario del registro de recursos anterior.
- ✚ **TTL: (Time To Live)** Tiempo de vida, indica el tiempo de vida durante el cual esa entrada puede ser considerada válida, es decir, el tiempo durante el cual se almacena esta entrada en la caché. Este campo es opcional. También se puede expresar mediante letras indicando días (d), horas (h), minutos (m) y segundos (s). Por ejemplo: "2h30m".
- ✚ **Clase:** define la familia de protocolos en uso. Suele ser siempre "IN", que representa Internet.
- ✚ **Tipo:** identifica el tipo de registro.
- ✚ **RDATA:** los datos del registro de recursos.

```
$TTL 86400
@      SOA      dns1.example.com.  hostmaster.example.com. (
        2001062501 ; serial
        21600      ; refresh after 6 hours
        3600       ; retry after 1 hour
        604800     ; expire after 1 week
        86400      ; minimum TTL of 1 day
)

;
;
; name servers - NS records
dns1   NS      dns1.example.com.
dns1   NS      dns2.example.com.
dns1   A       10.0.1.1
dns1   AAAA    aaaa:bbbb::1
dns2   A       10.0.1.2
dns2   AAAA    aaaa:bbbb::2
;
;
@      MX       10      mail.example.com.
@      MX       20      mail2.example.com.
mail   A       10.0.1.5
mail   AAAA    aaaa:bbbb::5
mail2  A       10.0.1.6
mail2  AAAA    aaaa:bbbb::6
;

$TTL 604800
@      IN       SOA      ns1.nyc3.example.com.  admin.nyc3.example.com. (
        3          ; Serial
        604800     ; Refresh
        86400      ; Retry
        2419200    ; Expire
        604800     ; Negative Cache TTL
)

;
; name servers - NS records
IN     NS       ns1.nyc3.example.com.
IN     NS       ns2.nyc3.example.com.

; name servers - A records
ns1.nyc3.example.com.  IN  A      10.128.10.11
ns2.nyc3.example.com.  IN  A      10.128.20.12

; 10.128.0.0/16 - A records
host1.nyc3.example.com.  IN  A      10.128.100.101
host2.nyc3.example.com.  IN  A      10.128.200.102
```

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

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

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

- ✚ **SOA**, siglas en inglés de Start Of Authority (Inicio de autoridad). Este es el primer registro de la zona y sólo puede haber uno en cada archivo de la zona y sólo está presente si el servidor es autoritario del dominio. Especifica el servidor DNS primario del dominio, la cuenta de correo del administrador y tiempo de refresco de los servidores secundarios.

- ✚ **NS**, siglas en inglés de Name Server (Nombre de servidor). Indica los servidores de DNS autorizados (principales y secundarios) para el dominio. Debe haber, al menos, uno.
- ✚ **A**, siglas en inglés de Address (Dirección). Es el registro más usado ya que permite enlazar un nombre de dominio o subdominio hacia una dirección IPv4.
- ✚ **Registro AAAA y A6**. Son muy similares al registro A, pero en lugar de apuntar a una dirección IPv4 apunta a una IPv6. No obstante, el registro A6 aún está en fase experimental por lo que de momento se recomienda el uso de AAAA:
- ✚ **PTR**, siglas en inglés de PoinTeR (Puntero). Son usados principalmente para la resolución inversa de nombres.
- ✚ **CNAME**, siglas en inglés de Canonical NAME (Nombre canónico). También se conoce como **Registro Alias**, permite asignar otro nombre de dominio, a un host que tiene una dirección IP válida, y que por lo tanto responderá a diversos nombres. Pueden declararse varios para un host
- ✚ **MX**, siglas en inglés de Mail Exchange (Intercambio de correo). Especifica el servidor de email responsable de distribuir los emails para tu dominio. Podemos tener más de uno, debiendo establecer una prioridad.

7.1.1 Tipo de registro SOA.

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

- ✚ **Propietario**: nombre de dominio de la zona.
- ✚ **Tipo**: "SOA".
- ✚ **Persona responsable**: contiene la dirección de correo electrónico del responsable de la zona. En esta dirección de correo se utiliza un punto en el lugar del símbolo "@".
- ✚ **Número de serie**: muestra el número de

```
$TTL      604800
@         IN      SOA      ns1.nyc3.example.com. admin.nyc3.example.com. (
                                3           ; Serial
                                604800      ; Refresh
                                86400       ; Retry
                                2419200     ; Expire
                                604800 )    ; Negative Cache TTL
;
```

- versión de la zona, es decir, un número que sirve de referencia a los servidores secundarios de la zona para saber cuándo deben proceder a una actualización de su base de datos de la zona (o transferencia de zona). Cuando el número de serie del servidor secundario sea menor que el número del maestro, esto significa que el maestro ha cambiado la zona, y por tanto el secundario debe solicitar al maestro una transferencia de zona. Por tanto, este número debe ser incrementado (manualmente) por el administrador de la zona cada vez que realiza un cambio en algún registro de la zona (en el servidor maestro).
- ✚ **Actualización (Refresh)**: muestra cada cuánto tiempo un servidor secundario debe ponerse en contacto con el maestro para comprobar si ha habido cambios en la zona, y por tanto pedir una **transferencia de zona**.
- ✚ **Reintentos (Retry)**: Tiempo que espera un servidor de nombres secundario para iniciar una nueva transferencia de zona en el caso de que falle este procedimiento.
- ✚ **Caducidad (Expire)**: define el tiempo que el servidor secundario de la zona, después de la transferencia de zona anterior, responderá a las consultas de la zona antes de descartar la suya propia como no válida.
- ✚ **Negative Cache TTL**: Número de segundos que los registros se mantienen activos en los servidores NS caché antes de volver a preguntar su valor real.

Un ejemplo de registro SOA sería el siguiente:

```
admon.com. IN      SOA      pc0100.admon.com hostmaster.admon.com. (
    2020110300 ; Serial Number (número de serie)
    1d12h      ; Refresh (Actualización) - 1día + 12 horas
    15m        ; Retry (Reintento) - 15 minutos
    3w12h      ; Expiry (Caducidad) - 3 semanas + 12 horas
    2h20m      ; Negative Cache TTL 2horas + 20 min
)
```

7.1.2 Tipo de registro NS

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

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

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

Ejemplos de registros NS serían los siguientes:

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

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

7.1.3 Tipo de registro A, PTR, CNAME Y MX.

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

Registro de Recurso A

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

```
; FQDN
pc0101.admon.com.      IN      A      158.42.178.1

; Relativo
pc0101                  IN      A      158.42.178.1
```

```
; name servers - A records
ns1.nyc3.example.com.  IN      A      10.128.10.11
ns2.nyc3.example.com.  IN      A      10.128.20.12

; 10.128.0.0/16 - A records
host1.nyc3.example.com. IN      A      10.128.100.101
host2.nyc3.example.com. IN      A      10.128.200.102
```

Registro de Recurso PTR

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

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

; Relativo
1.178                  IN      PTR      pc0101.admon.com.
```

```
; PTR Records
11.10 IN      PTR      ns1.nyc3.example.com. ; 10.128.10.11
12.20 IN      PTR      ns2.nyc3.example.com. ; 10.128.20.12
101.100 IN     PTR      host1.nyc3.example.com. ; 10.128.100.101
102.200 IN     PTR      host2.nyc3.example.com. ; 10.128.200.102
```

Registro de Recurso CNAME

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

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

Registro de Recurso MX

El registro de recurso de intercambio de correo (MX) especifica un servidor de intercambio de correo para un nombre de dominio. Puesto que un mismo dominio puede contener diferentes servidores de correo, el registro MX puede indicar un

valor numérico que permite especificar el orden en que los clientes deben intentar contactar con dichos servidores de correo.

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

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

8 Actividades

- 1) Utilizando el comando ping, ¿puedes averiguar la IP del nombre de dominio: www.iesdonana.org?
- 2) La utilidad nslookup nombre_dominio fuerza a la resolución del nombro de dominio especificado en el comando. Ejecuta nslookup www.yahoo.es y respondo a las siguientes preguntas:
 - a) ¿Qué significan las dos primeras líneas de salida?
 - b) ¿Qué significan las restantes?