UD 5. SERVICIOS DE RED IMPLICADOS EN EL DESPLIEGUE DE UNA APLICACIÓN WEB.

1. **SISTEMA DE NOMBRES**

En una red TCP/IP, las máquinas se identifican mediante su dirección de red o número IP. Sin embargo es mucho más fácil recordar un nombre de dominio que una sucesión de números.

Si alguien nos pide que visitemos la página 216.642.10.28 a los pocos segundos ni la recordamos, sin embargo si nos solicita entrar en cocacola.com lo recordaremos mucho mejor.

Hace años en 1983 los servidores eran tan pocos que cada host tenía una lista de **nombres de dominio - ip** y la actualizaban diariamente descargándose la información actualizada de un servidor que centralizaba esta información. Esto era un **sistema de nombres plano**.

En un espacio de nombres planos, todos los nombres deben ser absolutamente únicos: no puede haber 2 máquinas con el mismo nombre.

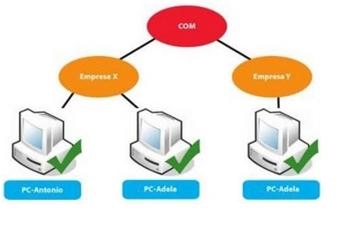
Para organizaciones grandes, esto no sirve, pues podría haber conflictos de nombres y sería imposible poder manejar los más de 50.000 nombres nuevos diarios.



SOLUCIÓN: SISTEMA DE NOMBRES JERÁRQUICO Y DISTRIBUIDO

El sistema de nombres jerárquico, tiene estructura de árbol de forma que cada nodo del árbol tiene un significado. Por ejemplo el espacio de nombres de los ficheros en disco (se pueden crear ficheros con el mismo nombre siempre que estén en otra carpeta)

1. disco\carpeta\subcarpeta\+...+nombre.



El sistema de nombres de dominio (Domain Name System) se encarga de esta gestión, traduce un nombre en una dirección IP, consultando en los servidores DNS.

**Sistema de Nombres de Dominio (Domain Name System)**

Desarrollado en noviembre de 1983 por Paul Mockapetris.

Se encarga de convertir direcciones formadas por cadenas de caracteres llamadas nombres DNS en direcciones IP, que serían más difíciles de recordar por los usuarios.

Podemos hacer referencia a los ordenadores a través de nombres DNS que lógicamente serán traducidos por “alguien” a direcciones IP que, al final, son lo único que entienden los sistemas TCP/IP. Ese alguien es un ordenador que ejecuta un programa servidor DNS y que, por tanto, se llama SERVIDOR DNS.

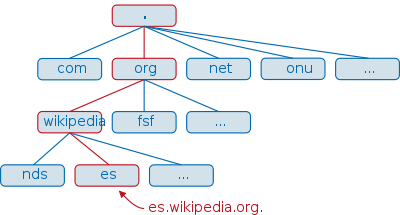
El **sistema de nombres de dominio** es un sistema de nombres jerárquico y distribuido. La estructura del sistema de nombres de dominio en una estructura en forma de árbol. Cada nodo del árbol se llama **nombre de dominio** yconsiste en una etiqueta de hasta 63 caracteres.

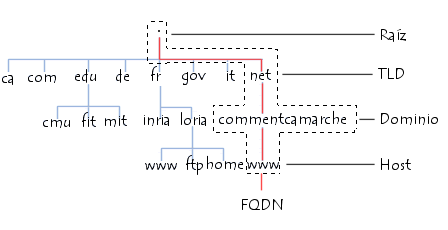
La raíz del árbol siempre es un PUNTO.

A la etiqueta de segundo nivel se le llama **dominio de nivel superior** o **Top Level Domain (TLD).**

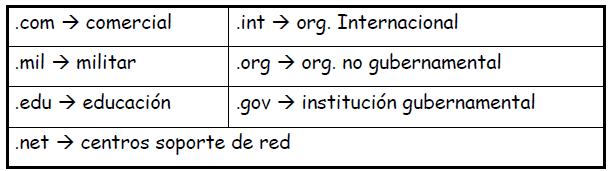
El extremo de la bifurcación se denomina **host** y corresponde a un equipo en la red.

**Una dirección FQDN** (*Nombre de Dominio totalmente calificado*) corresponde con todas las etiquetas de nodo de una estructura arbórea, separadas por puntos y que termina con un punto final. La profundidad máxima de una estructura arbórea es 127 niveles y la longitud máxima para un nombre FQDN es 255 caracteres.





El **Centro de Información de la Red Internet** ([Internet Network Information Center](http://www.internic.net/)) administra la raíz de la base de datos DNS en Internet. Los dominios superiores se han asignado a **organizaciones** y **países**. Estos nombres de dominio siguen un estándar internacional. Para los países se usan abreviaturas de dos y de tres letras (es, it, de…), y se han reservado varias abreviaturas para que las usen las organizaciones, como se muestra en los siguientes ejemplos:



1. **RESOLUTORES DE NOMBRES.**

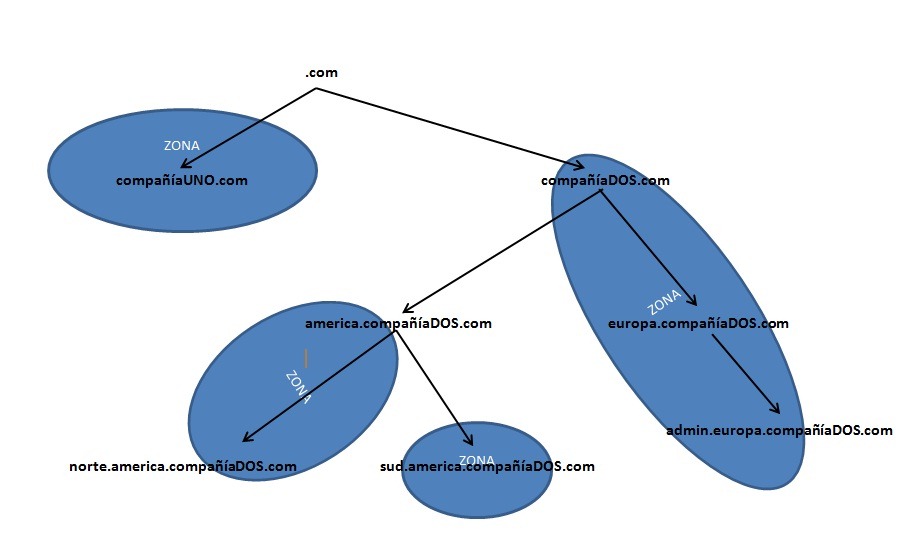
Cada nodo del árbol de una base de datos DNS, junto con todos los nodos por debajo del mismo, se llama dominio.

Los dominios pueden contener host (equipos) y otros dominios (subdominios).

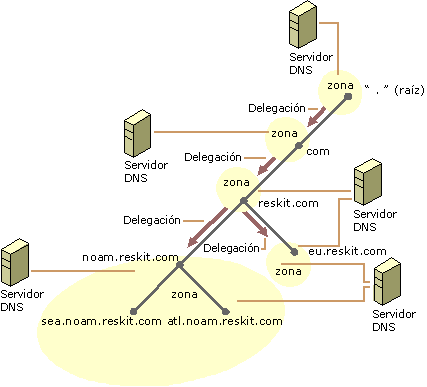
Por ejemplo, el dominio Midominio (midominio.com), podría contener a la vez equipos, como host.midominio.com, y subdominios, como subdom.midominio.com, que a su vez podría contener host, como por ejemplo host.subdom.midominio.com.

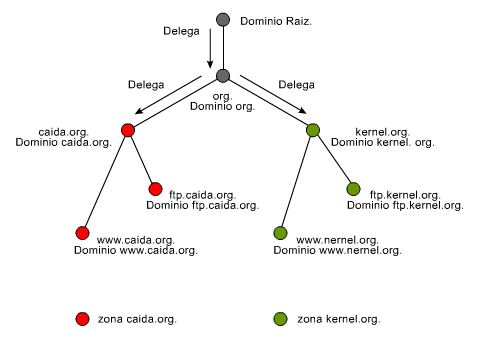
Existen equipos que actúan como **servidores de nombres de dominio** que permiten establecer la relación entre los nombres de dominio y las direcciones IP de los equipos de una red.

El sistema de servidores de nombres de dominio es una **arquitectura jeráquica y distribuida**, y permite su administración descentralizada mediante la **delegación de dominios**. Es decir como no habría un ordenador tan potente como para traducir todos los nombres de todos los ordenadores de Internet, lo que se hizo es que existen varios servidores DNS que trabajan de forma jerárquica y delegándose tareas unos en otros. Cada servidor DNS almacena información acerca de algunas partes del espacio de nombres del dominio. Cada una de esas partes se llama **zona.** Se dice el servidor DNS tiene **autoridad sobre una o varias zonas.**

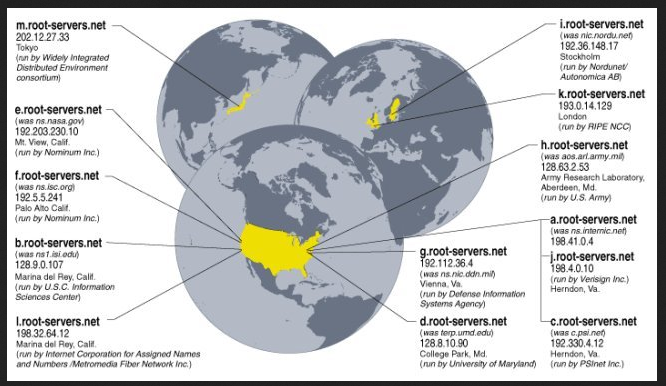


Por otra parte cada servidor de nombre de dominio está especificado en el servidor de nombre de dominio en el nivel superior inmediato, lo que significa que la autoridad sobre los dominios puede delegarse implícitamente, a esto se le llama **delegación de dominios, es decir un dominio puede ser dividido en subdominios delegando el control de otras zonas a otro servidor.**





Los servidores relacionados con los dominios de nivel superior (TLD) se llaman "**servidores de dominio de nivel superior**". Son 13, están distribuidos por todo el mundo y sus nombres van desde "a.root-servers.net" hasta "m.root-servers.net".



1. **Clasificación de tipos de servidores dns**

Hay 3 tipos de servidores DNS.

**Servidores primarios (maestros)**

La información de una zona la obtienen de sus archivos locales. Todas las modificaciones sobre una zona (añadir dominios, modificarlos, etc) se llevan a cabo en el servidor primario.

**Servidores secundarios (esclavos)**

Contiene una copia de sólo lectura de los archivos de zona de los servidores maestros con autoridad sobre esas zonas.

Cuando la información del servidor primario cambia, el esclavo la copia para actualizarse. Al proceso mediante el cual se obtiene información actualizada de la zona a través de internet se le llama “transferencia de zona”.

**Servidores caché**

Cuando se les hace una consulta, si no tiene la respuesta, la reenvían a los servidores que puedan responder. Al recibir la respuesta la almacenan en caché para poder usarla en sucesivas peticiones.

Si se almacena la información en el servidor caché por mucho tiempo, la info puede estar desactualizada, pero si se almacena muy poco tiempo, mayor carga para el sevidor de nombres. Hay que mirar lo que sea mejor para cada caso. Esto es el TTL o Time To Live (Tiempo de vida).

1. **PROCESO DE RESOLUCIÓN DE UN NOMBRE DE DOMINIO**

Cuando escribimos una dirección Web, el navegador debe resolver ese nombre en la dirección IP del servidor Web para establecer una conexión con ese servidor. Puede resolver el nombre de 2 formas:

**Consulta a fichero de texto**

Cuando un programa necesita resolver un nombre, normalmente, su primera parada suele ser el fichero HOSTS teniendo éste precedencia sobre DNS. En este fichero es donde comprueba si aparece o no una entrada para el nombre DNS especificado.

Windows 95/98: C:\Windows\hosts –

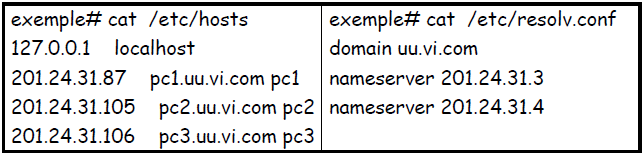
Windows NT\2000: C:\WINNT\System32\drivers\etc\hosts

Windows XP, 2003, Vista : C:\WINDOWS\System32\drivers\etc\hosts

Linux y Unix: /etc/hosts

MacOS: /etc/hosts

Este fichero contiene una línea por cada equipo que tenga asociado un nombre dns.

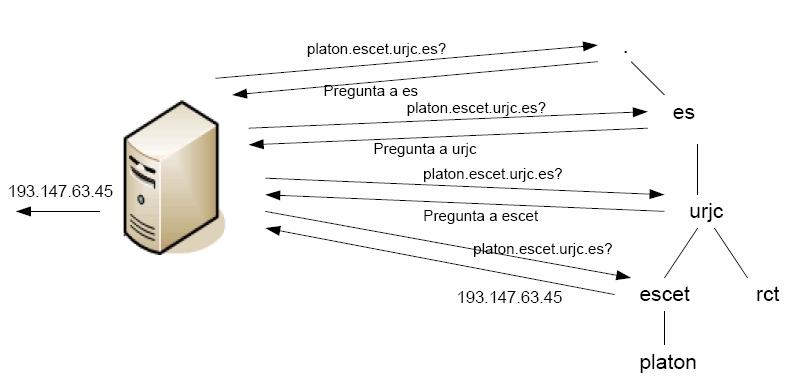
****

**Contacto con un servidor DNS**

Cuando una aplicación (cliente) necesita resolver un FQDN envía un requerimiento al servidor de nombres configurado en el sistema (normalmente, el provisto por el ISP). A partir de entonces se desencadena el proceso de resolución del nombre. Existen 2 mecanismos:

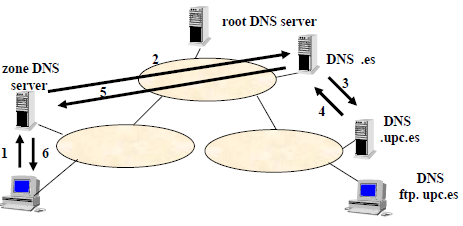
**Resolución iterativa**

1. El servidor de nombres inicial consulta a uno de los servidores raíz. (Actualmente existen 13 servidores raíz especificados, con los nombres de la forma letra.root-servers.net, donde letra va desde la A a la M.
2. Este devuelve el nombre del servidor a quien se le ha delegado la sub-zona.
3. El servidor inicial interroga al nuevo servidor.
4. El proceso se repite nuevamente a partir del punto 2 si es que se trata de una sub-zona delegada.
5. Al obtener el nombre del servidor con autoridad sobre la zona en cuestión, el servidor inicial lo interroga.
6. El servidor resuelve el nombre correspondiente, si este existe.
7. El servidor inicial informa al cliente el nombre resuelto.

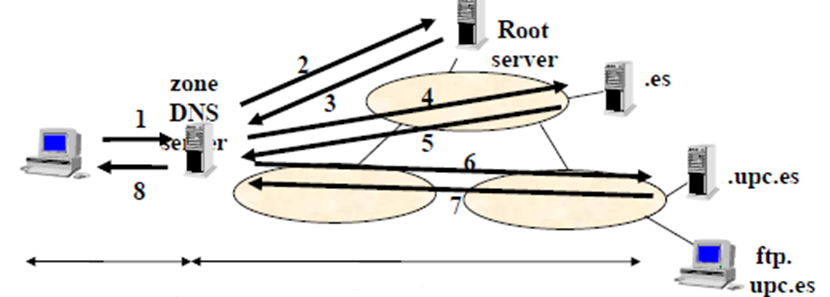


**Resolución recursiva**

1. (1) - Host pregunta por ftp.upc.es al servidor DNS de su zona (dominio)
2. (2) - El servidor DNS de la zona pregunta al DNS server con dominio .es
3. (3) - El servidor DNS .es pregunta al servidor DNS con dominio .upc.es
4. (4) - El servidor DNS .upc.es le devuelve la @IP del servidor ftp.upc.es al dominio .es
5. (5) (6) - Se devuelve la @IP del servidor ftp.upc.es al cliente.



**EJERCICIO:** Dada la siguiente imagen escribir qué pasa en cada paso. ¿Es iterativa o recursiva?



**EJERCICIO:** El comando “nslookup” sirve para hacer consultas DNS. Funciona en Windows, MAC o Linux. Usando la propia ayuda del comando descubre cómo se realiza: una consulta directa, una inversa, una consulta a un servidor DNS concreto (no al por defecto).

**Consulta directa**

**nslookup www.cpilosenlaces.com**

**Consulta inversa**

**nslookup 37.46.73.249**

**Consulta a un servidor dns en concreto (no al de por defecto)**

*root@ubuntu:/etc/bind# nslookup*

*> server d.in-addr-servers.arpa.*

*> www.google.es*

*Server: d.in-addr-servers.arpa.*

*Address: 200.10.60.53#53*

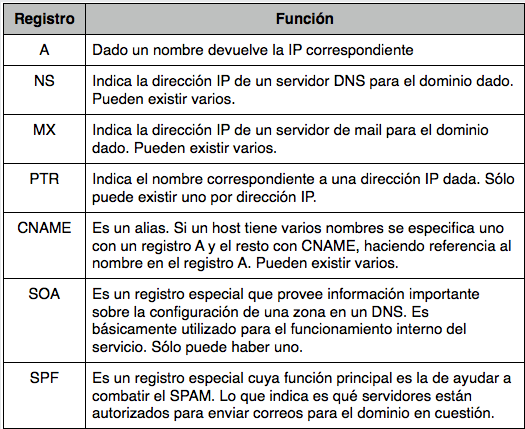
*\*\* server can't find www.google.es: REFUSED*

1. **Parámetros de configuración y registros del servidor de nombres afectados en el despliegue.**

**Los archivos del DNS.**

Un archivo de base de datos del servidor de nombres (o archivo DNS) es un ‘archivo de zona’ que contiene los registros (líneas del archivo) para la parte del dominio de la que es responsable la zona.

Los principales tipos de registros son:



Registro SOA

Registro NS

Registro MX

Registro A

Registro CNAME

Registro PTR

Registros SPF

**Registro SOA.**

El primer registro de recursos de cualquier archivo de zona del sistema de nombres de dominio (DNS) debe ser un registro de recursos del inicio de autoridad (SOA) **Start Of Authority.** Este registro indica la dirección del **servidor primario de esa zona** y datos relativos a la forma en la que se sincronizan los secundarios con el primario. Para cada zona,  debe existir un solo registro SOA.

Proporciona la siguiente información

* **host origen:** El host en el que se mantiene el archivo.
* **correo electrónico de contacto:** Dirección de correo electrónico de la persona responsable de administrar el archivo de zona del dominio. Hay que tener en cuenta que un "." se utiliza en lugar de una "@" en el nombre de correo electrónico.
* **número de serie:** Es el número de la revisión de este archivo de zona. Incrementar este número cada vez que se cambie el archivo de zona. Es importante incrementar este valor cada vez que se realice un cambio, para que los cambios se distribuyan a los servidores DNS secundarios.
* **tiempo de actualización:** El Tiempo, en segundos que un servidor DNS secundario espera antes de consultar el registro SOA del servidor DNS principal para comprobar los cambios. Cuando caduca el tiempo de actualización, el servidor DNS secundario solicita una copia del registro SOA actual del primario. El servidor DNS principal cumple con esta solicitud. El servidor DNS secundario compara el número de serie del registro SOA actual del servidor DNS principal y el número de serie en su propio registro SOA. Si son diferentes, el servidor DNS secundario solicitará una transferencia de zona desde el servidor DNS principal. El valor predeterminado es 3.600.
* **tiempo de reintento:** El tiempo antes de volver a intentar una transferencia de zona fallida. Normalmente, el tiempo de reintento es menor que el tiempo de actualización. El valor predeterminado es 600.
* **tiempo de expiración o caducidad:** El tiempo en segundos, que un servidor secundario seguirá intentando completar una transferencia de zona. Si este tiempo transcurre antes de una transferencia de zona exitosa, el servidor secundario caducará su archivo de zona. Esto significa que el secundario dejaría de responder las consultas, ya que considera sus datos demasiado antiguos como para ser confiables. El valor predeterminado es 86.400.
* **tiempo de vida (TTL):** el valor mínimo de tiempo de vida se aplica a todos los registros de recursos en el archivo de zona. Este valor se proporciona en respuestas a consultas para informar a otros servidores de cuánto tiempo deben mantener los datos en caché. El valor predeterminado es 3.600.

**Ejemplo:**

; El @ hace referencia al nombre de la zona

@ IN SOA nameserver.place.dom. postmaster.place.dom. (

1 ; serial number

3600 ; refresh [1h]

600 ; retry [10m]

86400 ; expire [1d]

3600 ) ; min TTL [1h]

**Ejemplo:**

mi-empresa–sa.es. IN SOA serv1.mi-empresa-sa.es. Jaime.mi-empresa-sa.es. (

2009082801; Numero serie (serial)

86400;  Actualización (refresh)

7200; Reintento (retry)

3600000; Expiración (expire)

172800); TTL Negativo

**Registro NS**

Cada uno de ellos (puede haber mas de uno) indica el FQDN de uno de los servidores de dominio de la zona. El formato es muy simple:

<nombre\_zona> IN NS <FQND\_Servidor>

Deben existir tantos registros NS como servidores de nombres hay para esa zona. El nombre que aparezca a la derecha será el que identifica a cada uno de los servidores. Estos servidores pueden estar dentro de la misma zona o fuera de esta.

midom.es. IN NS ns1.midom.es.

subdom.midom.es. IN NS ns.subdom.midom.es.

**Registro A**

Un Registro de dirección A (Address) sirve para asociar nombres de host a direcciones IP dentro de una zona. Éstos son los registros que componen la mayor parte del archivo de base de datos. Su formato es el siguiente nombrehost IN A direcciónIPdehost

Ejemplo

machine1 IN A 157.55.201.143

nombreservidor2 IN A 157.55.200.2

**El Registro CNAME**

Estos registros también reciben el nombre de alias, aunque son conocidos como entradas de "nombre canónico”. La utilidad principal de los mismos es la de usar más de un nombre para apuntar a un único host. Esto puede simplificar operaciones como albergar a la vez un servidor FTP y un servidor web en el mismo equipo. Su formato es el siguiente:

nombrealiashost IN CNAME nombrehost

Ejemplo: Suponga que servidorarchivos y que ftp se encuentran en el mismo equipo. Si éste es el caso, entonces podría tener las siguientes entradas en su archivo de zona:

servidorarchivos IN A 157.55.200.41

ftp IN CNAME servidorarchivos

www IN CNAME servidorarchivos

**REGISTROS PTR**

Los registros 'PTR' o apuntadores, se usan para la resolución inversa. También conocido como 'registro inverso', funciona a la inversa del registro A, traduciendo IPs en nombres de dominio.

Por ejemplo:

5.1.168.192.in-addr.arpa. IN PTR pc01.daw1.com.

6.1.168.192.in-addr.arpa. IN PTR pc02.daw2.com.

El siguiente comando (dig -x) se utiliza para preguntar por el nombre de una máquina dada la ip.

Ejemplo $ dig -x 127.0.0.1

;; ANSWER SECTION: 1.0.0.127.in-addr.arpa. 0 IN PTR localhost.

**Registro MX (Mail eXchange record)**

El Registro de Intercambio de Correo indica qué host procesa el correo de este dominio. Si existen múltiples registros de intercambio de correo, el resolvedor de nombres intentará ponerse en contacto con los servidores de correo en orden de preferencia, empezando por los valores inferiores (mayor prioridad) hasta el valor superior (menor prioridad). Su formato es el siguiente:

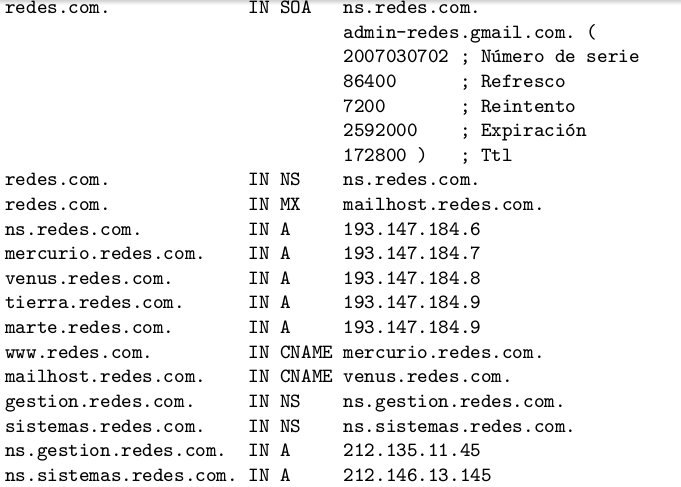
dominio IN MX preferencia servidorcorreohost

Ejemplo:

netstorming.com.ar. 7200 IN      MX      5 netstorming.com.ar.  
netstorming.com.ar.     7200    IN      MX     0 mail.netstorming.com.ar.

El resultado que se ve es que NetStorming tiene dos servidores de mail que son mail.netstorming.com.ar y netstorming.com.ar. El primero de ellos tiene mayor prioridad (de allí el 0).

**EJEMPLO COMPLETO DE ARCHIVO DE DATOS DE LA ZONA redes.com.**

****

1. **INSTALACIÓN Y configuración del servidor de nombres afectados en el despliegue.**

**Instalación del servidor BIND9 y configuración de zona**

Prácticamente el único software utilizado en los servidores de nombres de Internet es [bind](http://es.wikipedia.org/wiki/BIND) (“Berkeley Internet Name Domain“), creado originalmente en la Universidad de California, y actualmente propiedad del Internet Systems Consortium.

BIND es el servidor de nombres de dominio más popular en Internet, que trabaja en todas las plataformas informáticas principales y se caracteriza por su flexibilidad y seguridad.

Al contar con un **DNS** propio (ya sea uno o varios servidores de nombres) es posible definir *zonas locales* (no válidas ni accesibles desde *Internet*) para asignar nombres a cada uno de los *hosts* de la **LAN**. De esta forma es posible, por ejemplo, referirnos a la impresora de red como “*hplaser.mired.local*” en vez de “*192.168.0.2*” y a nuestro servidor de correo interno como “*smtp.mired.local*” en vez de “*192.168.0.3*“.

sudo apt-get install bind9

Vamos a configurar el archivo /etc/bind/named.conf.local donde declararemos las zonas.

El dominio que tenemos delegado se va a llamar despliegue.com. El archivo debería quedar de la siguiente forma:

zone "despliegue.com" {

type master;

file "/etc/bind/db.despliegue";

};

Explicación:

**Línea 1**: nombre de la zona

**Línea 2**: tipo de servidor: La línea type master significa que el dns que estamos configurando es primario. Podría ser secundario o esclavo (slave), o caché (hint).

**Línea 3:** hace referencia a la ruta del archivo donde se guardarán los registros de los nombres de dominio. Se recomienda que el fichero comience por db para hacer ver que se refiere a una base de datos.

Comprobamos que no hay ningún error de sintaxis. Para ello nos situamos en /etc/bind y hacemos:

named-checkconf

Si no pone nada es que está bien sintácticamente (no faltan “;”, “}” y cosas así)

El siguiente paso es modificar el archivo de configuración de DNS. En este paso se va a crear el fichero que va a almacenar la base de datos del servidor DNS, el nombre de dominio, la dirección IP, etc.

La ruta donde se va a crear es en este caso /etc/bind/ y el fichero se va a llamar db.despliegue tal y como hemos indicado en el anterior apartado.

$ttl 38400

despliegue.com. IN SOA ns.despliegue.com. admin.despliegue.com. ( 1317165042 10800 3600 604800 38400) despliegue.com. IN NS ns.despliegue.com.ns.despliegue.com. IN A 192.168.1.101pc01.despliegue.com. IN A 192.168.1.2pc02.despliegue.com. IN A 192.168.1.6www.despliegue.com. IN CNAME ns.despliegue.com.ftp.despliegue.com. IN CNAME ns.despliegue.com.

Podemos comprobar la correcta sintaxis con la orden

named-checkzone despliegue.com /etc/bind/db.despliegue

zone despliegue.com/IN: loaded serial 1317165042

OK

Donde el primer parámetro es el nombre de la zona y el segundo la ruta al archivo correspondiente.

Si todo ha ido bien (en la salida del comando nos dice OK) reiniciamos el server

sudo /etc/init.d/bind9 restart

Ahora hay que configurar el dns a utilizar que será el nuestro. Configuramos en la interfaz de red nuestro equipo como dns. Para ello en /etc/network/interfaces, **dns-nameservers ip\_servidor\_dns** (en vez de editar el fichero /etc/resolv.conf) Ahora ya no necesitamos el fichero /etc/hosts para resolver direcciones.

Comprobamos si resuelve las direcciones con los comandos nslookup, dig, y host.

nslookup ns.despliegue.com

nslookup [www.despliegue.com](http://www.despliegue.com)

dig www.despliegue.com

; <<>> DiG 9.9.5-3-Ubuntu <<>> www.despliegue.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48640

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;www.despliegue.com. IN A

;; ANSWER SECTION:

www.despliegue.com. 38400 IN CNAME ns.despliegue.com.

ns.despliegue.com. 38400 IN A 192.168.1.101

;; AUTHORITY SECTION:

despliegue.com. 38400 IN NS ns.despliegue.com.

;; Query time: 60 msec

;; SERVER: 192.168.1.101#53(192.168.1.101)

;; WHEN: Sun Feb 08 13:36:58 CET 2015

;; MSG SIZE rcvd: 94

Por otro lado si queremos resolver nombres de fuera de nuestra red y reenviar las peticiones a otro dns tenemos que configurar los forwarders. (Si no se configuran también se resuelven nombres fuera de nuestra red preguntando por defecto a los servidores raíz)

Lo que debemos hacer ahora es abrir el archivo de opciones.

gedit /etc/bind/named.conf.options

en él debemos quitar los comentarios de la parte de forwarders y configurarlo con el dns del isp o los que se usen para fuera de nuestra red.

forwarders {

8.8.8.8;

};

Si algo falla mirar /var/log/syslog

**Ejercicio: Probad otros comandos**

dig @servidorDns nombreDominio (Pregunta a un servidor por defecto)

dig +trace nombre (Traza de servidores dns consultados en la resolución)

ejemplo: Usando el programa dig, obtén la secuencia de servidores DNS que se sigue para resolver la dirección www.microsoft.com

dig +trace [www.microsoft.com](http://www.microsoft.com)

dig –t soa dominio (pregunta sobre el registro soa de un dominio)

dig –t ns dominio (pregunta sobre el registro ns de un dominio)

**Configuración de zona inversa**

Para poder preguntar por el nombre de dominio a partir de la ip necesitamos configurar la resolución inversa.

Vamos a configurar el fichero named.conf.local para la resolución inversa, para ello hay que modificar el fichero para añadir (a continuación de la zona anterior) la zona inversa identificada como la IP de la red al revés y sin la parte del host acabada en .in-addr.arpa.

// Definición de la zona inversa para DNS.

zone "1.168.192.in-addr.arpa" {

type master;

file "/etc/bind/db.192.168.1.in-addr.arpa";

};

Podemos comprobar que la sintaxis sea correcta con la orden que ya vimos. Si es correcta no aparecerá ningún mensaje pero si no lo es nos mostrará el problema.

named-checkconf

El siguiente paso es crear el archivo de zona inversa.

gedit /etc/bind/db.192.168.1.in-addr.arpa

$ttl 38400

@ IN SOA ns.despliegue.com. admin.despliegue.com. (

1317165042

10800

3600

604800

38400)

@ IN NS ns.despliegue.com.

101 IN PTR ns.despliegue.com.

2 IN PTR pc01.despliegue.com.

6 IN PTR pc02.despliegue.com.

Y lo comprobamos

named-checkzone 1.168.192.in-addr.arpa /etc/bind/db.192.168.1.in-addr.arpa

La @ se puede sustituir por el dominio 1.168.192.in-addr.arpa

Reiniciamos bind

/etc/init.d/bind9 restart

Debemos probar que la resolución inversa funciona correctamente.

root@ubuntu:/etc/bind# dig -x 192.168.1.101

; <<>> DiG 9.9.5-3-Ubuntu <<>> -x 192.168.1.101

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64829

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;101.1.168.192.in-addr.arpa. IN PTR

;; ANSWER SECTION:

101.1.168.192.in-addr.arpa. 38400 IN PTR ns.despliegue.com.

;; AUTHORITY SECTION:

1.168.192.in-addr.arpa. 38400 IN NS ns.despliegue.com.

;; ADDITIONAL SECTION:

ns.despliegue.com. 38400 IN A 192.168.1.101

;; Query time: 40 msec

;; SERVER: 192.168.1.101#53(192.168.1.101)

;; WHEN: Sun Feb 08 14:03:27 CET 2015

;; MSG SIZE rcvd: 116

EJERCICIO: Pon 2 máquinas virtuales en puente, en una tendrás instalado bind y vsftpd y tomcat.

1. Comprueba que desde la máquina cliente y utilizando el dns configurado anteriormente puedes acceder al servidor ftp mediante la orden ftp [ftp.despliegue.com](ftp://ftp.despliegue.com).
2. Configura un sitio virtual en tomcat que se llame tomcat.despliegue.com e intenta acceder desde el cliente

EJERCICIO: Si tuvieras también Apache con 2 sitios virtuales basados en nombres [www.despliegue.com](http://www.despliegue.com) y [www.despliegue2.com](http://www.despliegue2.com) ¿Cómo configurarías bind9 para poder gestionar los 2 dominios?

**SOLUCIÓN**

Como despliegue2 es otro dominio, necesitaremos

1. Crear una nueva zona despliegue2.com. dentro de /etc/bind/named.conf.local
2. Crear el archivo de zona
3. Añadir dentro de este archivo los datos del despliegue2 con un registro CNAME apuntando al primero es decir crear un registro para el nuevo dominio pero realmente será un alias al anterior.