



Universidad Adventista de Centro América

Escuela de Ingeniería

Tema:
DNS

DOMAIN NAME SYSTEM



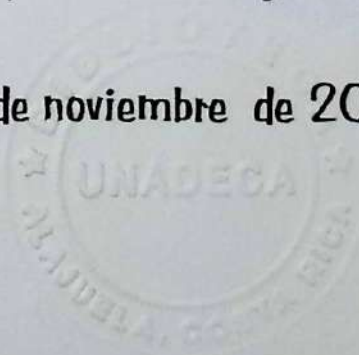
Trabajo Presentado a:
Lic. Jair del Valle

En cumplimiento parcial de la materia:
TELEMÁTICA Y REDES

Por:
Carlos A. Aquino López

39.455

Alajuela, Costa Rica, 9 de noviembre de 2000



INTRODUCCIÓN

El presente material ha sido una recopilación de información obtenidas de algunas paginas web de Internet, en las cuales se busco todo lo relacionado con el tema de Domain Name System, o Sistemas de Nombres de Dominio.

Hacemos notar que este material es solo una pequeña muestra de la basta información que en Internet se encuentra.

Se espera de que sea una pequeña guía y despierte el interés del lector para que continúe con la investigación del Tema y pueda aportar más.

El este presente manual no se incluye una Bibliografía para referenciarse a las paginas web visitadas para la recopilación de este tema. Si el lector desea visitar alguna de estas paginas, podrá darse cuenta de que las direcciones se encuentran debajo de cada tema Principal.

Esperamos que este manual pueda serte de mucha ayuda.

INDICE

| | |
|---|-----------|
| Introducción | 01 |
| Servidor de Nombre de “Solo Cacheo” | 03 |
| Arranque de named | 07 |
| Un Dominio Simple | 09 |
| Nuestro Propio Dominio | 13 |
| Un Ejemplo de Dominio Real | 22 |
| Mantenimiento | 29 |
| Configuración de Conexiones Automáticas vía Telefónica | 30 |
| Preguntas de Uso Frecuente (FAQ) | 33 |
| Anexo | 37 |
| Cronología del DNS | 39 |
| NIC-Internet Costa Rica Información y Dominio | 43 |
| Como Instalar un Servidor de DNS | 51 |
| El Software | 52 |
| El Archivo de Arranque | 53 |
| Los Archivos de Zonas | 58 |
| Verificando la Instalación | 66 |
| Registros de Configuración DNS | 72 |
| Servicio DNS | 78 |
| Servidor DNS | 79 |
| Intratec | 80 |
| Configuración Para DHCP | 81 |

Domain Name System (Sistema de Nombres de Dominio),

1. Introducción.

Lo que es esto y lo que no es.

Para los que comienzan, DNS es el Domain Name System (Sistema de Nombres de Dominio), las reglas de nomenclatura de las máquinas y el software que mapea los nombres a números IP. Este documento COMO trata de cómo definir tales conversiones usando un sistema Linux. Una conversión es simplemente una asociación entre dos cosas, en este caso un nombre de máquina, como ftp.linux.org y el número IP de la máquina, 199.249.150.4.

El DNS es, para los no iniciados (Vd. :-), una de las áreas mas opacas de la administración de una red. Este COMO tratará de aclarar algunas cosas.

Este documento describe cómo configurar un servidor de nombres DNS simple. Comenzaremos con un servidor *caching only server* Servidor que se limita a guardar en una caché las IPs de los nombres de máquina más solicitados, obteniéndolas de servidores externos. , y continuaremos con la configuración de un servidor DNS primario para un dominio. Para configuraciones más complejas puede consultar la sección de PUF (PUF) de este documento. Si lo que busca no está descrito allí, necesitará *Documentación Real*. Volveremos a lo que es la *Documentación Real* en el último capítulo.

Antes de empezar, debe configurar su sistema convenientemente, de forma que pueda hacer telnet desde y hacia su máquina, efectuando satisfactoriamente toda clase de conexiones de red, especialmente telnet 127.0.0.1 entrando en su propia máquina (compruébelo ahora). También necesita que los archivos /etc/host.conf (o /etc/nsswitch.conf), /etc/resolv.conf y /etc/hosts sean correctos como punto de partida, ya que no explicaré sus funciones aquí. Si NO tiene aun esta configuración y no funciona en red, el *NET-2 HOWTO* explica como hacerlo. Léalo.

Si está usando SLIP o PPP necesitará que funcionen correctamente. Lea el *PPP-COMO* si no es así.

Cuando digo "su máquina" quiero decir la máquina en la que está intentando configurar DNS. No cualquier otra máquina que pueda tener en su red.

Supongo que no está detrás de cualquier clase de cortafuegos (*firewall*) que bloquee peticiones de nombres. Si necesita una configuración especial, vea la sección PUF (PUF).

El servicio de nombres en Unix es llevado a cabo por un programa, llamado `named`.

Éste forma parte del paquete `bind`, que es coordinado por Paul Vixie para *The Internet Software Consortium*. `named` está incluido en la mayoría de las distribuciones de Linux y generalmente se instala como `/usr/sbin/named`. Si tiene el fichero `named` probablemente pueda usarlo; si no lo tiene, puede obtener el binario en un ftp de Linux, o conseguir los últimos y más voluminosos fuentes en <ftp://ftp.vix.com/pub/bind>, bien de los subdirectorios de versión actual, o de prueba, lo que mejor se adapte a su estilo de vida.

DNS es una base de datos cuyo ámbito es la Red. Tenga cuidado con lo que pone en ella. Si pone incongruencias, Vd. y los demás obtendrán incongruencias de ella. Mantenga su DNS limpia y consistente y conseguirá un buen servicio de ella. Aprenda a usarla, administrarla, depurarla y será otro buen administrador, salvando a la red de caer sobre sus rodillas sobrecargada por falta de mantenimiento.

En este documento expongo de forma llana varias cosas que no son completamente verdad (son al menos medias verdades). Todo esto lo hago en aras de la simplicidad. Todas funcionarán (probablemente ;-) si cree en lo que digo.

Aviso:

Haga una copia de seguridad de todos los archivos que le indico que cambie si ya los tiene, y así si después si no funciona podrá volver al principio.

2. Servidor de nombres de "sólo cacheo".

Un primer ataque a la configuración DNS, muy útil para los usuarios de conexiones telefónicas.

Un servidor de nombres de "sólo cacheo" (*caching only nameserver*) obtendrá la respuesta a las solicitudes de nombre provenientes de su red preguntando a servidores externos, recordando la respuesta para la próxima vez que lo necesite.

Lo primero que necesita es el archivo llamado `/etc/named.boot`. Este archivo es leído cuando se inicia `named`. Por ahora contendrá simplemente:

Archivo boot de servidor de nombres de solo cacheo:

```
directory /var/named
```

| tipo | dominio | fichero o maquina fuente |
|---------|----------------------|--------------------------|
| cache | . | root.cache |
| primary | 0.0.127.in-addr.arpa | pz/127.0.0 |

MUY IMPORTANTE:

En algunas versiones de este documento, en el contenido de los archivos que aquí aparecen hay un par de espacios o tabuladores antes del primer carácter no blanco. Se supone que estos caracteres NO están en el archivo. Borre cualquier espacio inicial de los archivos que corte y pegue de este COMO.

La línea `directory` indica a `named` dónde buscar los archivos. Todos los archivos indicados a continuación serán relativos a este directorio. `/var/named` es el directorio correcto de acuerdo con el *LFS, Linux File system Standard*. Así, `pz` es un directorio bajo `/var/named`, esto es, `/var/named/pz`.

3.1 /var/named/root.cache

Vamos a describir el archivo llamado `/var/named/root.cache` nombrado en el archivo `boot.named`.

`/var/named/root.cache` debería contener esto:

```
. 518400 NS D.ROOT-SERVERS.NET.  
. 518400 NS E.ROOT-SERVERS.NET.  
. 518400 NS I.ROOT-SERVERS.NET.  
. 518400 NS F.ROOT-SERVERS.NET.  
. 518400 NS G.ROOT-SERVERS.NET.  
. 518400 NS A.ROOT-SERVERS.NET.  
. 518400 NS H.ROOT-SERVERS.NET.  
. 518400 NS B.ROOT-SERVERS.NET.  
. 518400 NS C.ROOT-SERVERS.NET.
```

```
D.ROOT-SERVERS.NET. 3600000 A 128.8.10.90  
E.ROOT-SERVERS.NET. 3600000 A 192.203.230.10  
I.ROOT-SERVERS.NET. 3600000 A 192.36.148.17  
F.ROOT-SERVERS.NET. 3600000 A 192.5.5.241  
G.ROOT-SERVERS.NET. 3600000 A 192.112.36.4  
A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4  
H.ROOT-SERVERS.NET. 3600000 A 128.63.2.53  
B.ROOT-SERVERS.NET. 3600000 A 128.9.0.107  
C.ROOT-SERVERS.NET. 3600000 A 192.33.4.12
```

¡Recuerde lo que dije sobre los espacios iniciales!

Este archivo describe los servidores de nombres raíz en el mundo. Este archivo cambiará a lo largo del tiempo y tiene que ser mantenido y actualizado con una cierta regularidad. Vea la sección de mantenimiento ([mantenimiento](#)) para saber cómo mantenerlo actualizado. Este archivo está descrito en la página man de

named, pero esto es, IMHO *In My Honest Opinion*, EMMO o *En Mi Modesta Opinión* en castellano. , más apropiado para gente que ya comprende named.

La siguiente línea de named.boot es la línea primary. Explicaré su uso en un capítulo posterior: Por ahora, cree un archivo llamado 127.0.0 en el subdirectorio pz:

```
@      IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (  
      1      ; Numero de Serie  
      28800  ; Tasa de Refresco  
      7200   ; Tasa de Reintento  
      604800 ; Caducidad para secundario  
      86400) ; Validez para Clientes  
      NS      ns.linux.bogus.  
1      PTR      localhost.
```



A continuación necesita el archivo `/etc/resolv.conf`, que será algo similar a este:

```
search subdominio.su-dominio.edu su-dominio.edu  
nameserver 127.0.0.1
```

La línea `'search'` especifica en qué dominios se buscaría para cualquier nombre de máquina a la que quiera conectar. La línea `'nameserver'` especifica la dirección de su servidor de nombres, en este caso su propia máquina, ya que es ahí donde named se estará ejecutando. Si quiere una lista de varios servidores ponga una línea `nameserver` para cada uno. (Nota: named nunca lee este archivo, lo hace el resolutor que usa named).

Para ilustrar lo que hace este archivo:

Si un cliente intenta buscar a fulano, `fulano.subdominio.su-dominio.edu` se probará primero, a continuación `fulano.su-dominio.edu`, y finalmente `fulano`. Si un cliente intenta buscar `sunsite.unc.edu`, `sunsite.unc.edu.subdominio.su-dominio.edu` se prueba primero (sí, es tonto, pero es así como tiene que ser), después `sunsite.unc.edu.su-dominio.edu`, y finalmente `sunsite.unc.edu`. Puede que no quiera poner demasiados dominios en la línea `search`, lleva su tiempo el efectuar las búsquedas.

El ejemplo supone que pertenece al dominio subdominio.su-dominio.edu, su máquina probablemente se llame su-maquina.subdominio.su-dominio.edu. La línea search no debería contener su *TLD* (*Top Level Domain* o *Dominio de Nivel Superior*, 'edu' en este caso). Si necesita conectar frecuentemente con máquinas de otro dominio, puede añadir ese dominio a la línea search como sigue:

```
search subdominio.su-dominio.edu su-dominio.edu otro-dominio.com
```

y así sucesivamente. Obviamente necesita poner un dominio real en su lugar. Por favor, dése cuenta de la falta de puntos al final de estos nombres de dominio.

Lo siguiente, dependiendo de su versión de la librería libc puede necesitar arreglar `/etc/nsswitch.conf` o `/etc/host.conf`. Si ya tiene `nsswitch.conf` corregiremos éste, en otro caso arreglaremos `host.conf`.

2.2 `/etc/nsswitch.conf`

Se trata de un extenso archivo donde se especifica de dónde obtener las diferentes clases de tipos de datos, y de cuál archivo o base de datos. Generalmente contiene comentarios útiles al comienzo, que por cierto debería considerar leer ahora. Después busque la línea que comienza por `hosts:`, debe leerse:

```
hosts: files dns
```

Si no hay una línea que comience por 'hosts:' póngala. Eso indica que los programas deben mirar primero en el fichero `/etc/hosts`, y después comprobar DNS de acuerdo con `resolv.conf`.

2.3 `/etc/host.conf`

Probablemente contiene varias líneas, una de ellas debería comenzar con `order` y tendría que parecerse a lo siguiente:

```
order hosts,bind
```

Si no hay una línea `order` tiene que incluirla. Esto le indica a las rutinas de resolución de nombres que busquen primero en `/etc/hosts`, y pregunte luego al servidor de nombres (que dijo en `resolv.conf` que está en `127.0.0.1`). Estos dos últimos archivos están documentados en la página de manual `resolv(8)` (haciendo `man 8 resolv`) en la mayoría de las distribuciones Linux. Esta página de manual es de obligada lectura *IMHO*, y todos, especialmente los administradores DNS, deberían leerla. Hágalo ahora, si se dice a sí mismo *"lo haré más tarde"* entonces nunca lo hará.

2.4 Arranque de `named`

Después de todo esto, ya es hora de iniciar `named`. Si está utilizando una conexión telefónica, conéctese primero. Teclee `ndc start` y presione `return`, sin opciones. Si tiene problemas intente `/usr/sbin/ndc start` en su lugar. Si el problema persiste vea la sección `PUF` (`PUF`). Ahora ya puede comprobar su configuración. Si mira en el archivo de mensajes de `syslog` (generalmente llamado `/var/adm/messages`, o en el directorio `/var/log`) mientras inicia `named`, (haga `tail -f /var/adm/messages`) debería ver algo como esto:

```
Jun 30 21:50:55 roke named[2258]: starting.  named 4.9.4-REL Sun Jun 30
21:29:03 MET DST 1996
      janl@roke.slip.ifi.uio.no:/var/tmp/bind/named
Jun 30 21:50:55 roke named[2258]: cache zone "" loaded (serial 0)
Jun 30 21:50:55 roke named[2258]: primary zone "0.0.127.in-addr.arpa" loaded
(serial 1)
```

Si hay cualquier mensaje de error se deberá a alguna equivocación. `named` determinará el archivo que ocasiona el error (de `named.boot` o `root.cache` espero :-). Mate a `named` y vuelva a comprobar el archivo.

Ahora es el momento de iniciar `nslookup` para examinar su trabajo:

```
$ nslookup
Default Server: localhost
```

Address: 127.0.0.1

Si es eso lo que obtiene entonces está funcionando. Eso espero. En cualquier otro caso, vuelva atrás y compruébelo todo. Cada vez que cambie el archivo `named.boot` tiene que reinicializar `named` usando el comando `ndc restart`.

Ahora puede introducir una consulta. Intente buscar alguna máquina cercana a la suya. `pat.uio.no` está cerca de mí, en la Universidad de Oslo:

```
> pat.uio.no
Server: localhost
Address: 127.0.0.1
```

```
Name: pat.uio.no
Address: 129.240.2.50
```

`nslookup` ahora solicita a `named` que busque la máquina `pat.uio.no`. Contactará con alguna de las máquinas servidoras de nombres nombradas en el archivo `root.cache`, y preguntará allí. Puede tardar un poco antes de conseguir el resultado ya que busca todos los dominios indicados en `/etc/resolv.conf`.

Si intenta de nuevo obtendrá esto:

```
> pat.uio.no
Server: localhost
Address: 127.0.0.1
```

```
Non-authoritative answer:
Name: pat.uio.no
Address: 129.240.2.50
```

Nota a la línea ``Non-authoritative answer:" : le dedicaremos un poco de tiempo. Esto significa que `named` no sale de la red para preguntar esta vez, en su lugar mira en su caché y lo encuentra allí. Pero la información de la caché puede no estar actualizada. Entonces informa de este peligro (de modo un tanto eufemístico) con `Non-authorative answer:`. Cuando `nslookup` dice esto la segunda vez que pregunta por una máquina, es un signo seguro de que `named` almacena la información en la caché y que está funcionando. Ahora puede salir de `nslookup` usando el comando `exit`.

Si es un usuario de conexiones telefónicas, (ppp, slip) por favor lea la sección sobre conexiones telefónicas (timofonicas), hay algunas advertencias para Vd. Ahora ya sabe cómo configurar un servidor de nombres de "sólo cacheo". Tómese una cerveza, un vaso de leche o cualquier otra cosa que prefiera para celebrarlo.

3. Un dominio simple.

Como configurar su propio dominio.

3.1 Pero primero algo de teoría a secas.

Antes de comenzar realmente con esta sección, voy a dar un poco de teoría sobre cómo funciona DNS. Y lo va a leer porque será mejor para Vd. Si no quiere, al menos debería echar un vistazo rápido. Deje el repaso cuando sepa lo que debe incluir en su archivo `named.boot`.

El DNS es un sistema jerárquico. La raíz se escribe como `.` y se denomina *'root'*. Debajo hay cierto número de Dominios de Nivel Superior (*Top Level Domains, TLDs*), los más conocidos son ORG, COM, EDU y NET, pero hay muchos más.

Cuando se busca una máquina, la pregunta procede recursivamente en la jerarquía comenzando desde arriba. Si quiere localizar la dirección de `prep.ai.mit.edu`, su servidor de nombres ha de encontrar primero un servidor de nombres que sirva a `edu`. Pregunta al servidor `.` (ya conoce los servidores `.`, es para lo que se utiliza el archivo `root.cache`), y el servidor `.` proporcionará una lista de servidores `edu`:

```
$ nslookup
Default Server: localhost
Address: 127.0.0.1
```

Comienza preguntando a un servidor raíz.

```
> server c.root-servers.net.
```

Default Server: c.root-servers.net

Address: 192.33.4.12

Pone el tipo de petición (*Query*) a NS (*Name Server records*).

> set q=ns

Pregunta por edu.

> edu.

El punto (".") final aquí es significativo, indica al servidor que le pedimos un edu que está justo debajo de ".", y esto reduce la búsqueda un poco.

```
edu  nameserver = A.ROOT-SERVERS.NET
edu  nameserver = H.ROOT-SERVERS.NET
edu  nameserver = B.ROOT-SERVERS.NET
edu  nameserver = C.ROOT-SERVERS.NET
edu  nameserver = D.ROOT-SERVERS.NET
edu  nameserver = E.ROOT-SERVERS.NET
edu  nameserver = I.ROOT-SERVERS.NET
edu  nameserver = F.ROOT-SERVERS.NET
edu  nameserver = G.ROOT-SERVERS.NET
A.ROOT-SERVERS.NET  internet address = 198.41.0.4
H.ROOT-SERVERS.NET  internet address = 128.63.2.53
B.ROOT-SERVERS.NET  internet address = 128.9.0.107
C.ROOT-SERVERS.NET  internet address = 192.33.4.12
D.ROOT-SERVERS.NET  internet address = 128.8.10.90
E.ROOT-SERVERS.NET  internet address = 192.203.230.10
I.ROOT-SERVERS.NET  internet address = 192.36.148.17
F.ROOT-SERVERS.NET  internet address = 192.5.5.241
G.ROOT-SERVERS.NET  internet address = 192.112.36.4
```

Esto nos dice que *.root-servers.net sirve a edu., y así podemos seguir preguntando a C. Ahora queremos saber quién sirve el siguiente nivel del nombre de dominio: mit.edu.

> mit.edu.

Server: c.root-servers.net

Address: 192.33.4.12

Non-authoritative answer:

mit.edu nameserver = STRAWB.mit.edu



mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu

Authoritative answers can be found from:

STRAWB.mit.edu internet address = 18.71.0.151

W20NS.mit.edu internet address = 18.70.0.160

BITSY.mit.edu internet address = 18.72.0.3

steawb, w20ns y bitsy sirven a mit, selecciona uno y pregunta por ai.mit.edu:

> server W20NS.mit.edu.

Los nombres de máquina no son sensibles a mayúsculas/minúsculas, pero como yo uso el ratón para cortar y pegar, obtengo una copia tal y como aparece en la pantalla.

Server: W20NS.mit.edu

Address: 18.70.0.160

> ai.mit.edu.

Server: W20NS.mit.edu

Address: 18.70.0.160

Non-authoritative answer:

ai.mit.edu nameserver = WHEATIES.AI.MIT.EDU

ai.mit.edu nameserver = ALPHA-BITS.AI.MIT.EDU

ai.mit.edu nameserver = GRAPE-NUTS.AI.MIT.EDU

ai.mit.edu nameserver = TRIX.AI.MIT.EDU

ai.mit.edu nameserver = MUESLI.AI.MIT.EDU

Authoritative answers can be found from:

AI.MIT.EDU nameserver = WHEATIES.AI.MIT.EDU

AI.MIT.EDU nameserver = ALPHA-BITS.AI.MIT.EDU

AI.MIT.EDU nameserver = GRAPE-NUTS.AI.MIT.EDU

AI.MIT.EDU nameserver = TRIX.AI.MIT.EDU

AI.MIT.EDU nameserver = MUESLI.AI.MIT.EDU

WHEATIES.AI.MIT.EDU internet address = 128.52.32.13

WHEATIES.AI.MIT.EDU internet address = 128.52.35.13

ALPHA-BITS.AI.MIT.EDU internet address = 128.52.32.5

ALPHA-BITS.AI.MIT.EDU internet address = 128.52.37.5

GRAPE-NUTS.AI.MIT.EDU internet address = 128.52.32.4

GRAPE-NUTS.AI.MIT.EDU internet address = 128.52.36.4

TRIX.AI.MIT.EDU internet address = 128.52.32.6

TRIX.AI.MIT.EDU internet address = 128.52.38.6

MUESLI.AI.MIT.EDU internet address = 128.52.32.7

MUESLI.AI.MIT.EDU internet address = 128.52.39.7

Entonces `wheaties.ai.mit.edu` es un servidor de nombres para `ai.mit.edu`:

```
> server WHEATIES.AI.MIT.EDU.
```

```
Default Server: WHEATIES.AI.MIT.EDU
```

```
Addresses: 128.52.32.13, 128.52.35.13
```

Ahora cambia el tipo de solicitud; ha encontrado el servidor de nombres y va a preguntar todo lo que queremos saber sobre `prep.ai.mit.edu`.

```
> set q=any
```

```
> prep.ai.mit.edu.
```

```
Server: WHEATIES.AI.MIT.EDU
```

```
Addresses: 128.52.32.13, 128.52.35.13
```

```
prep.ai.mit.edu CPU = dec/decstation-5000.25 OS = unix
```

```
prep.ai.mit.edu
```

```
inet address = 18.159.0.42, protocol = tcp
```

```
#21 #23 #25 #79
```

```
prep.ai.mit.edu preference = 1, mail exchanger = life.ai.mit.edu
```

```
prep.ai.mit.edu internet address = 18.159.0.42
```

```
ai.mit.edu nameserver = alpha-bits.ai.mit.edu
```

```
ai.mit.edu nameserver = wheaties.ai.mit.edu
```

```
ai.mit.edu nameserver = grape-nuts.ai.mit.edu
```

```
ai.mit.edu nameserver = mini-wheats.ai.mit.edu
```

```
ai.mit.edu nameserver = trix.ai.mit.edu
```

```
ai.mit.edu nameserver = muesli.ai.mit.edu
```

```
ai.mit.edu nameserver = count-chocula.ai.mit.edu
```

```
ai.mit.edu nameserver = life.ai.mit.edu
```

```
ai.mit.edu nameserver = mintaka.lcs.mit.edu
```

```
life.ai.mit.edu internet address = 128.52.32.80
```

```
alpha-bits.ai.mit.edu internet address = 128.52.32.5
```

```
wheaties.ai.mit.edu internet address = 128.52.35.13
```

```
wheaties.ai.mit.edu internet address = 128.52.32.13
```

```
grape-nuts.ai.mit.edu internet address = 128.52.36.4
grape-nuts.ai.mit.edu internet address = 128.52.32.4
mini-wheats.ai.mit.edu internet address = 128.52.32.11
mini-wheats.ai.mit.edu internet address = 128.52.54.11
mintaka.lcs.mit.edu internet address = 18.26.0.36
```

De esta forma comenzando en . ha encontrado los sucesivos servidores de nombre para el siguiente nivel en el nombre de dominio. Si ha usado su propio servidor DNS en lugar de usar todos esos otros servidores, su `named`, desde luego, habrá almacenado en el caché toda la información que haya encontrado mientras profundizaba en la búsqueda, y en consecuencia no tendrá que preguntar de nuevo durante un tiempo.

Se habla mucho menos sobre él, pero un dominio importante es `in-addr.arpa`. También está anidado como los dominios '*normales*'. `in-addr.arpa` nos permite determinar el nombre de la máquina cuando conocemos su dirección IP. Una cosa importante aquí es observar que que las direcciones IP están escritas en orden inverso en el dominio `in-addr.arpa`. Si tiene la dirección de máquina `192.128.52.43`, `named` procede como para el ejemplo de `prep.ai.mit.edu`: Busca los servidores `arpa.`. Busca los servidores `in-addr.arpa.`, los servidores `192.in-addr.arpa.`, los servidores `128.192.in-addr.arpa.` , y los servidores `52.128.192.in-addr.arpa.` y finalmente, los registros necesarios para `43.52.128.192.in-addr.arpa.` ¿Inteligente? (Diga `sí'). La inversión de números puede ser confusa los 2 primeros años. He contado una mentira. DNS no funciona como he dicho de forma literal. Pero es bastante parecido.

3.2 Nuestro propio dominio

Ahora vamos a definir nuestro propio dominio. Vamos a crear el dominio `linux.bogus` y definir máquinas en él. Uso un nombre de dominio totalmente falso para estar seguro de que no molestamos a nadie de fuera.

Ya hemos comenzado esta parte con la siguiente línea en named.boot:

```
primary 0.0.127.in-addr.arpa pz/127.0.0
```

Por favor tome nota de la ausencia de `.' al final de los nombres de dominio en este archivo. La primera línea nombra al archivo pz/127.0.0 como definición de 0.0.127.in-addr.arpa. Ya hemos configurado este archivo, en él podremos leer:

```
@      IN      SOA    ns.linux.bogus. hostmaster.linux.bogus. (
        1      ; Numero de Serie
        28800 ; Tasa de Refresco
        7200  ; Tasa de Reintento
        604800 ; Caducidad para secundario
        86400) ; Tiempo de Validez para Clientes
        NS    ns.linux.bogus.
1      PTR    localhost.
```

Por favor observe los `.' al final de los nombres de dominio completo en contraste con el archivo named.boot anterior. A algunas personas les gusta iniciar cada zona del archivo con una directiva \$ORIGIN, pero esto es superfluo. El origen (lugar de la jerarquía DNS a donde pertenece) de un fichero de zona se especifica en la columna dominio del archivo named.boot; en este caso es 0.0.127.in-addr.arpa.

Este "fichero de zona" contiene tres *registros de recursos* (RRs): Un RR SOA, Un RR NS y un RR PTR. SOA es una abreviatura de *Start Of Authority*. La `@' es una notación especial que simboliza el origen, y como la columna dominio para este archivo indica 0.0.127.in-addr.arpa. La primera línea realmente significa:

```
0.0.127.IN-ADDR.ARPA. IN      SOA ...
```

NS es el RR Name Server (Servidor de Nombres), e indica a DNS qué máquina es el servidor de nombres del dominio. Y finalmente el registro PTR tiene valor 1 (igual a 1.0.0.127.IN-ADDR.ARPA, esto es, 127.0.0.1) que es el localhost de named.

El registro SOA es el preámbulo de todos los archivos de zona y debe haber uno exactamente en cada archivo de zona, como primer registro de todos. El registro

SOA describe la zona, de dónde proviene (una máquina llamada linux.bogus), quién es el responsable de su contenido (hostmaster@linux.bogus), qué versión del archivo de zona es (Número de Serie, 1), y otras cosas que tienen que ver con el caché y los servidores secundarios DNS. Para el resto de los campos (Tasa de Refresco, Tasa de Reintento, Caducidad para secundario y Tiempo de Validez para Clientes) use los valores que aparecen aquí para mayor seguridad.

El registro NS nos indica quién efectúa el servicio DNS para 0.0.127.in-addr.arpa, que es ns.linux.bogus. El registro PTR nos dice que 1.0.0.127.in-addr.arpa (aka 127.0.0.1) es conocido como localhost.

Ahora reiniciamos named (el comando es `ndc restart`) y usamos `nslookup` para examinar lo que ha hecho:

```
$ nslookup
```

```
Default Server: localhost
```

```
Address: 127.0.0.1
```

```
> 127.0.0.1
```

```
Server: localhost
```

```
Address: 127.0.0.1
```

```
Name: localhost
```

```
Address: 127.0.0.1
```

así obtiene localhost de 127.0.0.1, bien. Ahora para nuestra tarea principal, el dominio linux.bogus, inserte una nueva línea, primary, en named.boot:

```
primary      linux.bogus      pz/linux.bogus
```

Observe que continúa la ausencia de "." final en el nombre de dominio del archivo named.boot.

En el archivo de zona de linux.bogus pondremos algunos datos totalmente falsos

```
N                                del                                T
```

Por si no lo ha notado todavía, *bogus* en inglés significa precisamente *falso*. :

```
;
```

```
; Fichero de zona para linux.bogus
```

```
;
```

```

; Minimo indispensable para tener funcionando un dominio
;
@ IN SOA ns.linux.bogus. hostmaster.linux.bogus. (
    199511301 ; Numero de serie, fecha de hoy + n. de serie de
    hoy
    28800 ; Tasa de Refresco, en segundos
    7200 ; Tasa de Reintento, en segundos
    3600000 ; Caducidad para secundario, en segundos
    86400 ) ; Tiempo de Validez para Clientes, en segundos
NS ns.linux.bogus.
NS ns.friend.bogus.
MX 10 mail.linux.bogus ; Intercambiador de Correo Primario
MX 20 mail.friend.bogus. ; Intercambiador de Correo Secundario

```

```

localhost A 127.0.0.1
ns A 127.0.0.2
mail A 127.0.0.4

```

Deben de observarse dos cosas sobre los registros SOA. ns.linux.bogus debe ser una máquina actual con un registro A. No es legal tener un registro CNAME para la máquina mencionada en el registro SOA. Su nombre no necesita ser ns, podría ser cualquier nombre legal de máquina. A continuación, en hostmaster.linux.bogus deberá aparecer algo como hostmaster@linux.bogus; esto sería un alias de email, o una cuenta de correo, donde la(s) persona(s) que realizan el mantenimiento de DNS deberían leer con frecuencia el correo. Cualquier email respecto del dominio será mandado a la dirección aquí indicada. El nombre no tiene por que ser hostmaster, puede ser cualquier dirección email legal, pero la dirección email hostmaster funcionará bien.

Hay un nuevo tipo de RR en este archivo, el MX, o *Mail eXchanger*. Este indica el sistema de correo a donde mandar el correo dirigido a alguien@linux.bogus, pudiendo ser también mail.linux.bogus o mail.friend.bogus. El número que precede a cada nombre de máquina es la prioridad del RR MX. El RR con el número más bajo (10) es aquel al que el correo será enviado primero. Si este falla, puede ser

mandado a otro con un número más alto, que será gestor secundario de correo, como mail.friend.bogus que tiene una prioridad 20 aquí.

Reinicie named ejecutando `ndc restart`. Examine los resultados con `nslookup`:

```
$ nslookup
> set q=any
> linux.bogus
Server: localhost
Address: 127.0.0.1

linux.bogus
  origin = linux.bogus
  mail addr = hostmaster.linux.bogus
  serial = 199511301
  refresh = 28800 (8 hours)
  retry = 7200 (2 hours)
  expire = 604800 (7 days)
  minimum ttl = 86400 (1 day)
linux.bogus  nameserver = ns.linux.bogus
linux.bogus  nameserver = ns.friend.bogus
linux.bogus  preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
linux.bogus  preference = 20, mail exchanger = mail.friend.bogus
linux.bogus  nameserver = ns.linux.bogus
linux.bogus  nameserver = ns.friend.bogus
ns.linux.bogus internet address = 127.0.0.2
mail.linux.bogus internet address = 127.0.0.4
```

Con un examen cuidadoso podrá descubrir un error. La línea

```
linux.bogus preference = 10, mail exchanger =
mail.linux.bogus.linux.bogus
```

está equivocada. Debería ser

```
linux.bogus preference = 10, mail exchanger = mail.linux.bogus
```

Cometí el error de forma deliberada para que aprenda de él :-). Mirando en el archivo de zona podemos ver que la línea

```
@ MX 10 mail.linux.bogus ; Intercambiador de Correo Primario
```

no tiene punto. O tiene demasiados linux.bogus. Si un nombre de máquina no termina en punto en un archivo de zona, el origen es añadido a su final. Así,

```
@ MX 10 mail linux.bogus. ; Intercambiador de Correo Primario
```

o

```
@ MX 10 mail ; Primary Mail Exchanger
```

serán correctos. Yo prefiero la última forma, hay que escribir menos. En un archivo de zona el dominio debería ser escrito y terminado con un punto, o no debe ser incluido, en cuyo caso se referirá al origen por defecto. Debo hacer hincapié que en el archivo named.boot no debería haber puntos después de los nombres de dominio. No tiene ni idea de cuantas veces un '.' por estar o por no estar ha hecho fallar toda una configuración y ha confundido horrorosamente a la gente...

Una vez hecha esta *puntualización*, he aquí el nuevo archivo de zona, con algo de información extra también:

```
;  
; Archivo de zona para linux.bogus  
;  
; minimo indispensable para hacer funcionar un dominio  
;  
@ IN SOA ns linux.bogus. hostmaster linux.bogus. (  
199511301 ; Numero de Serie, fecha de hoy + n. de serie de  
hoy  
28800 ; Tasa de Refresco, en segundos  
7200 ; Tasa de Reintento, en segundos  
604800 ; Caducidad para secundario, en segundos  
86400 ) ; Validez para Clientes, en segundos  
  
NS ns ; Direccion de Internet del servidor de nombres  
NS ns.friend.bogus.  
MX 10 mail ; Intercambiador de Correo Primario  
MX 20 mail.friend.bogus. ; Intercambiador de Correo Secundario  
  
localhost A 127.0.0.1  
ns A 127.0.0.2
```

```

mail      A      127.0.0.4
;
; Extras
;
@         TXT    "Linux.Bogus, your DNS consultants"

ns        MX     10 mail
          MX     20 mail.friend.bogus.
          HINFO  "Pentium" "Linux 1.2"
          TXT    "RMS"
richard   CNAME  ns
www       CNAME  ns

donald    A      127.0.0.3
          MX     10 mail
          MX     20 mail.friend.bogus.
          HINFO  "i486" "Linux 1.2"
          TXT    "DEK"

mail      MX     10 mail
          MX     20 mail.friend.bogus.
          HINFO  "386sx" "Linux 1.0.9"

ftp       A      127.0.0.5
          MX     10 mail
          MX     20 mail.friend.bogus.
          HINFO  "P6" "Linux 1.3.59"

```

Puede que quiera desplazar los tres primeros registros tipo A (localhost, ns y mail) junto con los otros registros de su mismo tipo (donald, mail, y ftp), en vez de colocarlos separados al principio como aquí.

Hay varios registros nuevos aquí: HINFO (*Host INFORMATION*), tiene dos partes, es una buena costumbre poner comillas a cada uno. La primera parte es el hardware o CPU de la máquina, y la segunda parte corresponde al software o Sistema Operativo de la misma. ns tiene una CPU Pentium con Linux 1.2. El

registro TXT es un texto en formato libre que puede usar para cualquier cosa que le interese. CNAME (*Canonical NAME*) es una forma de dar a cada máquina varios nombres. Por tanto richard y www son alias para ns. Es importante observar que los registros A, MX, CNAME y SOA nunca deben hacer referencia al registro CNAME, sólo pueden referirse a registros A.

```
fulanito CNAME richard ; !!NO!!!
```

siendo correcto tener

```
fulanito CNAME ns ; !!SI!!!
```

También es importante observar que CNAME no es un nombre de máquina legal para direcciones de correo: webmaster@www.linux.bogus es una dirección email ilegal dada en la configuración anterior. Encontrará muy pocos administradores de correo de Ahí Afuera que recomienden esta regla, incluso si a Vd. le funciona. La forma de evitar esto es usar un registro A (y quizás algunos otros también, como un registro MX) en su lugar:

```
www A 127.0.0.2
```

Paul Vixie, el principal gurú de named recomienda no usar CNAME. Por tanto considere el no utilizarlo seriamente.

Cargue la nueva base de datos ejecutando `ndc reload`, esto provoca la lectura de sus archivos de nuevo.

```
$ nslookup
Default Server: localhost
Address: 127.0.0.1
```

```
> ls -d linux.bogus
```

Esto haría que todos los registros fuesen listados.

```
[localhost]
linux.bogus. SOA ns.linux.bogus hostmaster.linux.bogus. (199511301
28800 7200 604800 86400)
linux.bogus. NS ns.linux.bogus
linux.bogus. NS ns.friend.bogus
linux.bogus. MX 10 mail.linux.bogus
linux.bogus. MX 20 mail.friend.bogus
linux.bogus. TXT "Linux.Bogus, your DNS consultants"
```

```

localhost      A  127.0.0.1
mail           A  127.0.0.4
mail           MX  10  mail.linux.bogus
mail           MX  20  mail.friend.bogus
mail           HINFO 386sx   Linux 1.0.9
donald         A  127.0.0.3
donald         MX  10  mail.linux.bogus
donald         MX  20  mail.friend.bogus
donald         HINFO i486   Linux 1.2
donald         TXT  "DEK"
www            CNAME ns.linux.bogus
richard        CNAME ns.linux.bogus
ftp            A  127.0.0.5
ftp            MX  10  mail.linux.bogus
ftp            MX  20  mail.friend.bogus
ftp            HINFO P6     Linux 1.3.59
ns             A  127.0.0.2
ns             MX  10  mail.linux.bogus
ns             MX  20  mail.friend.bogus
ns             HINFO Pentium Linux 1.2
ns             TXT  "RMS"
linux.bogus.   SOA  ns.linux.bogus hostmaster.linux.bogus. (199511301
28800 7200 604800 86400)

```

Esto está bien. Comprobemos qué dice para www sólo:

```

> set q=any
> www.linux.bogus.
Server: localhost
Address: 127.0.0.1

```

```
www.linux.bogus canonical name = ns.linux.bogus
```

...En otras palabras, el nombre real de www.linux.bogus es ns.linux.bogus

```

linux.bogus  nameserver = ns.linux.bogus
linux.bogus  nameserver = ns.friend.bogus
ns.linux.bogus internet address = 127.0.0.2

```

y ns.linux.bogus tiene la dirección 127.0.0.2. Parece correcto también.

3.3 Relajémonos

Desde luego, este dominio es falso, y como tal son todas sus direcciones, y quizás, desafortunadamente sea un poco confuso. Para un ejemplo real de dominio vea la siguiente sección.

4. Un ejemplo de dominio real

Donde describiremos algunos archivos de zona reales.

Los usuarios han sugerido que incluya un ejemplo real de dominio que esté en funcionamiento como explicación de las diferencias entre un dominio en funcionamiento y el ejemplo falso que no era del todo muy claro.

Una cosa sobre este ejemplo: ¡NO lo introduzca en su servidor de nombres!. Úselo sólo como lectura de referencia. Si quiere experimentar, hágalo con el ejemplo falso. Yo uso este ejemplo con permiso de David Bullock y LAND-5. Estos archivos eran los usados el 24 de Septiembre de 1996, y podrían diferir de los que encuentre si pregunta ahora al servidor de nombres LAND-5. También tenga en mente eliminar los espacios iniciales ;-).

4.1 `/etc/named.boot` (o `/var/named/named.boot`)

Aquí encontramos la líneas `primary` para las dos zonas que necesitamos: la red 127.0.0.0 y también la subred 206.6.177 de LAND-5. Una línea `primary` para la zona de redirección (`forward`) `land-5.com` de `land-5`. Observe también que en lugar de situar los archivos en un directorio llamado `pz`, como hago en este COMO, él los sitúa en un directorio llamado `zone`.

; Fichero de arranque para el servidor de nombres LAND-5

directory /var/named

| ; tipo | dominio | fichero o máquina origen |
|---------|------------------------|--------------------------|
| cache | . | root.cache |
| primary | 0.0.127.in-addr.arpa | zone/127.0.0 |
| primary | 177.6.206.in-addr.arpa | zone/206.6.177 |
| primary | land-5.com | zone/land-5.com |

4.2 /var/named/root.cache

Tenga en cuenta que este archivo varía con mucha frecuencia, y que el listado de aquí es viejo. Mejor utilice uno producido ahora.

```
; <<>> DiG 2.1 <<>>
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr rd ra; Ques: 1, Ans: 9, Auth: 0, Addit: 9
;; QUESTIONS:
;;    ., type = NS, class = IN

;; ANSWERS:
. 518357 NS    H.ROOT-SERVERS.NET.
. 518357 NS    B.ROOT-SERVERS.NET.
. 518357 NS    C.ROOT-SERVERS.NET.
. 518357 NS    D.ROOT-SERVERS.NET.
. 518357 NS    E.ROOT-SERVERS.NET.
. 518357 NS    I.ROOT-SERVERS.NET.
. 518357 NS    F.ROOT-SERVERS.NET.
. 518357 NS    G.ROOT-SERVERS.NET.
. 518357 NS    A.ROOT-SERVERS.NET.
```

:: ADDITIONAL RECORDS:

```
H.ROOT-SERVERS.NET. 165593 A 128.63.2.53
B.ROOT-SERVERS.NET. 165593 A 128.9.0.107
C.ROOT-SERVERS.NET. 222766 A 192.33.4.12
D.ROOT-SERVERS.NET. 165593 A 128.8.10.90
E.ROOT-SERVERS.NET. 165593 A 192.203.230.10
I.ROOT-SERVERS.NET. 165593 A 192.36.148.17
F.ROOT-SERVERS.NET. 299616 A 192.5.5.241
G.ROOT-SERVERS.NET. 165593 A 192.112.36.4
A.ROOT-SERVERS.NET. 165593 A 198.41.0.4
```

:: Total query time: 250 msec

:: FROM: land-5 to SERVER: default – 127.0.0.1

:: WHEN: Fri Sep 20 10:11:22 1996

:: MSG SIZE sent: 17 rcvd: 312

4.3 /var/named/zone/127.0.0

Lo básico, el registro obligatorio SOA, y el registro que mapea 127.0.0.1 a localhost. Se requieren ambos. No debería haber ninguno más en este fichero. Probablemente nunca se necesitará actualizarlo, salvo que cambien su servidor de nombres o la dirección del hostmaster.

```
@      IN      SOA     land-5.com. root.land-5.com. (
        199609203 ; Numero de Serie
        28800   ; Tasa de Refresco
        7200    ; Tasa de Reintento
        604800 ; Caducidad para secundario
        86400) ; Validez para clientes
NS     land-5.com.

1      PTR     localhost.
```

4.4 /var/named/zone/land-5.com

Aquí vemos el registro SOA y los registros NS necesarios. Podemos observar que dispone de un servidor de nombres secundario ns2.psi.net. Esto es como debe ser, tenga siempre un servidor secundario de seguridad. También podemos ver que tiene una máquina principal llamada land-5 que se encarga de todos los diferentes servicios, y que se ha hecho usando CNAME (una alternativa al uso de los registros A).

Como puede ver en el registro SOA, el origen del archivo de zona es land-5.com, la persona de contacto es root@land-5.com. hostmaster es otro uso frecuente para la persona de contacto. El número de serie en el formato habitual *yyyymmdd* con el número de serie de hoy añadido; esta es probablemente la sexta versión del archivo de zona del 20 de Septiembre de 1996. Recuerde que el número de serie debe incrementarse monótonamente, aquí hay sólo un dígito para las series de hoy, así que después de 9 ediciones tendrá que esperar hasta mañana antes de poder editar el el archivo de nuevo. Considere el uso de dos dígitos.

```
@    IN    SOA    land-5.com. root.land-5.com. (
                199609206      ; Numero de Serie, fecha de hoy + numero de
serie de hoy
                10800        ; Tasa de Refresco, en segundos
                7200         ; Tasa de Reintento, en segundos
                10800        ; Caducidad para secundario, en segundos
                86400 )      ; Validez para Clientes, en segundos
NS    land-5.com.
NS    ns2.psi.net.
MX    10 land-5.com. ; Intercambiador Primario de Correo

localhost    A    127.0.0.1

router       A    206.6.177.1

land-5.com.  A    206.6.177.2
ns          CNAME land-5.com.
```

ftp CNAME land-5.com.
www CNAME land-5.com.
mail CNAME land-5.com.
news CNAME land-5.com.

funn A 206.6.177.3
illusions CNAME funn.land-5.com.
@ TXT "LAND-5 Corporation"

; Estaciones de Trabajo

ws_177200 A 206.6.177.200
MX 10 land-5.com. ; Primary Mail Host
ws_177201 A 206.6.177.201
MX 10 land-5.com. ; Primary Mail Host
ws_177202 A 206.6.177.202
MX 10 land-5.com. ; Primary Mail Host
ws_177203 A 206.6.177.203
MX 10 land-5.com. ; Primary Mail Host
ws_177204 A 206.6.177.204
MX 10 land-5.com. ; Primary Mail Host
ws_177205 A 206.6.177.205
MX 10 land-5.com. ; Primary Mail Host
; {Muchas definiciones repetitivas borradas}
ws_177250 A 206.6.177.250
MX 10 land-5.com. ; Primary Mail Host
ws_177251 A 206.6.177.251
MX 10 land-5.com. ; Primary Mail Host
ws_177252 A 206.6.177.252
MX 10 land-5.com. ; Primary Mail Host
ws_177253 A 206.6.177.253
MX 10 land-5.com. ; Primary Mail Host
ws_177254 A 206.6.177.254
MX 10 land-5.com. ; Primary Mail Host

Otra cosa a tener en cuenta es que las estaciones de trabajo no tienen nombres propios, sino un prefijo seguido por las dos últimas porciones de los números IP. Usar tal convención puede simplificar el mantenimiento significativamente, pero puede resultar un poquito impersonal.

4.5 /var/named/zone/206.6.177

Comentaré este archivo después.

```
@      IN      SOA   land-5.com. root.land-5.com. (
        199609206 ; Numero de Serie
        28800   ; Tasa de Refresco
        7200   ; Tasa de Reintento
        604800 ; Caducidad para secundario
        86400) ; Validez para Clientes
      NS   land-5.com.
      NS   ns2.psi.net.
;
;   Servidores
;
1   PTR   router.land-5.com.
2   PTR   land-5.com.
3   PTR   funn.land-5.com.
;
;   Estaciones de Trabajo
;
200 PTR   ws_177200.land-5.com.
201 PTR   ws_177201.land-5.com.
202 PTR   ws_177202.land-5.com.
203 PTR   ws_177203.land-5.com.
204 PTR   ws_177204.land-5.com.
205 PTR   ws_177205.land-5.com.
; {Eliminadas muchas definiciones repetitivas}
250 PTR   ws_177250.land-5.com.
251 PTR   ws_177251.land-5.com.
252 PTR   ws_177252.land-5.com.
```

253 PTR ws_177253.land-5.com.

254 PTR ws_177254.land-5.com.

La zona de resolución inversa es la parte de la configuración que parece crear más dolores de cabeza. Se usa para encontrar el nombre de la máquina a partir de su dirección IP. Ejemplo: suponga que está en un servidor irc y acepta conexiones de clientes irc. El servidor irc es noruego y sólo quiere aceptar conexiones de clientes de Noruega y otros países escandinavos. Cuando se produce una conexión de un cliente, la librería *C* es capaz de indicar el número IP de la máquina conectada porque el número IP del cliente está contenido en todos los paquetes que se pasan a través de la red. Ahora puede llamar a una función llamada `gethostbyaddr` que busca el nombre de la máquina dada su dirección IP.

`gethostbyaddr` interrogará a un servidor DNS el cual efectuará una búsqueda DNS para la máquina. Suponiendo que la conexión cliente viene de `ws_177200.land-5.com`, la dirección IP que la librería *C* proporciona al servidor irc será `206.6.177.200`. Para encontrar el nombre de la máquina necesitamos encontrar `200.177.6.206.in-addr.arpa`. El servidor DNS primero encuentra los servidores `arpa.`, después los servidores `in-addr.arpa.`, a continuación sigue por `206`, `6` y al final busca el servidor para la zona `177.6.206.in-addr.arpa` en `land-5`. Aquí obtendrá finalmente que para `200.177.6.206.in-addr.arpa` tenemos un registro `'PTR ws_177200.land-5.com'`, que significa que el nombre que va con `206.6.177.200` es `ws_177200.land-5.com`. Como con la explicación de cómo buscar `prep.ai.mit.edu`, esto es ligeramente ficticio.

Volviendo al ejemplo del servidor irc. El servidor irc sólo acepta conexiones de los países escandinavos, osea, `*.no`, `*.se`, y `*.dk`; el nombre `ws_177200.land-5.com` claramente no se ajusta a ninguno de ellos, y el servidor denegará la conexión. Si no hubiese habido resolución inversa de `206.2.177.200` mediante la zona `in-addr.arpa` el servidor habría sido incapaz de encontrar el nombre y habría tenido que comparar `206.2.177.200` con `*.no`, `*.se` y `*.dk`, es decir, cifras con nombres, ninguna de las cuales concordaría.

Algunas personas le dirán que la resolución inversa sólo es importante para los servidores, o que no tienen importancia. No es así; muchos servidores de ftp, news, irc e incluso algunos servidores http (WWW) NO aceptarán conexiones de máquinas de las cuales no son capaces de resolver el nombre. Por tanto el mapeo inverso de máquinas es de hecho obligatorio

5. Mantenimiento

Manteniéndolo en funcionamiento.

Hay una tarea de mantenimiento que tiene que realizar con named, además de mantenerlo en funcionamiento. Esta tarea es mantener el archivo root.cache actualizado. La forma más fácil es usar dig, primero ejecute dig sin argumentos, conseguirá root.cache de acuerdo con su propio servidor. Entonces pregunte a alguno de los servidores raíz listados con

```
dig @rootserver
```

Podrá observar que la salida se parece terriblemente al archivo root.cache excepto por un par de números extras. Esos números no ocasionan problemas.

Guárdelo en un archivo

```
dig @rootserver . ns > root.cache.new
```

y sustituya el antiguo root.cache con él.

Recuerde reiniciar named después de sustituir el archivo caché.

Al Longyear me envió este script que puede ser ejecutado automáticamente para actualizar root.cache, instale una entrada en el crontab para ejecutarlo una vez al mes y olvídense. El script supone que trabaja con correo y que el alias de mail hostmaster está definido. Debe editarlo para ajustarlo a su configuración.

```
#!/bin/sh
#
# Actualizacion del cache del servidor de nombres una vez al mes.
# Esto es ejecutado automaticamente mediante una entrada de cron
#
(
```

```

echo "To: hostmaster <hostmaster>"
echo "From: system <root>"
echo "Subject: Actualizacion automatica del fichero named.boot"
echo
export PATH=/sbin:/usr/sbin:/bin:/usr/bin:
cd /var/named
dig @rs.internic.net . ns >root.cache.new

echo "El fichero named.boot ha sido actualizado para contener la
siguiente informacion:"
echo
cat root.cache.new

chown root.root root.cache.new
chmod 444 root.cache.new
rm -f root.cache.old
mv root.cache root.cache.old
mv root.cache.new root.cache
ndc restart
echo
echo "El servidor de nombres ha sido rearrancado a fin de asegurar que la
actualizacion es completa."
echo "El anterior fichero root.cache se ha renombrado a
/var/named/root.cache.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0

```

Alguno de ustedes puede haber observado que el archivo root.cache está también disponible mediante ftp en *Internic*. Por favor NO utilice ftp para actualizar root.cache, el método anterior es mucho más amistoso con la red.

6. Configuración de Conexiones Automáticas vía telefónica .

Esta sección explica cómo he dispuesto las cosas para automatizarlo todo. Mi método puede que no se adapte completamente al suyo, pero puede obtener ideas de algunas de las cosas que he hecho. También, uso ppp para marcar,

mientras que mucha gente usa `slip` o `cslip` y por tanto casi toda su configuración puede ser distinta a la mía. Pero el programa de `slip` `dip` debería poder hacer muchas de las cosas que yo hago.

Normalmente, cuando no estoy conectado a la red tengo un archivo `resolv.conf` que simplemente contiene la línea

```
domain uio.no
```

Eso me asegura que no tengo que esperar a que la librería de resolución de nombres del sistema intente conectar con un servidor de nombres que no puede ayudarme. Pero cuando me conecto quiero arrancar mi `named` y tener un `resolv.conf` parecido a los descritos anteriormente. He resuelto esto teniendo dos archivos `resolv.conf` llamados `resolv.conf.local` y `resolv.conf.connected`. El último se parece al `resolv.conf` descrito anteriormente en este documento.

Para conectarme automáticamente a la red ejecuto un script llamado `ppp-on`:

```
#!/bin/sh
echo llamando...
pppd
```

`pppd` tiene un archivo llamado `options` que indica las características de la conexión. Una vez que mi conexión `ppp` está activa `pppd` llama a un *script* llamado `ip-up` (este está descrito en la página `pppd` (8) de `man`). He aquí una parte del *script*:

```
#!/bin/sh
interface="$1"
device="$2"
speed="$3"
myip="$4"
upip="$5"
...
cp -v /etc/resolv.conf.connected /etc/resolv.conf
...
/usr/sbin/named
```

Es decir, arranco el `named` desde aquí. Cuando se corta la conexión `ppp`, `pppd` ejecuta un script llamado `ip-down`:

```
#!/bin/sh
cp /etc/resolv.conf.local /etc/resolv.conf
read namedpid < /var/run/named.pid
kill $namedpid
```

Así configuramos las cosas de una forma cuando estamos conectados y las desconfiguramos cuando nos desconectamos.

Algunos programas, irc y talk me vienen a la mente, hacen algunas suposiciones, y para que en irc el comportamiento de las capacidades dcc, y talk funcionen bien tiene que modificar su archivo hosts. Yo he insertado en mi script ip-up lo siguiente:

```
cp /etc/hosts.ppp /etc/hosts
echo $myip roke >>/etc/hosts
```

hosts.ppp simplemente contiene

```
127.0.0.1 localhost
```

y echo inserta la dirección IP que he recibido para mi nombre de host (roke). Vd. deberá usar en su lugar el nombre de su máquina. Este nombre se puede saber con el comando hostname.

Probablemente no sea inteligente ejecutar named cuando no esté conectado a la red, esto es porque named intentará enviar solicitudes a la red y eso consume tiempo, y Vd. tendrá que esperar este tiempo cada vez que algún programa intente resolver un nombre. Si está usando conexiones telefónicas debería iniciar named cuando se conecte y matarlo cuando se desconecte. Pero por favor lea la sección de PUF (PUF) para los trucos.

A algunas personas le gusta usar la directiva forwarders para conexiones de escasa velocidad. Si su proveedor de Internet tiene servidores DNS en 1.2.3.4 y 1.2.3.5 puede insertar la línea

```
forwarders 1.2.3.4 1.2.3.5
```

en el archivo named.boot. Deje también vacío el archivo root.cache. Esto disminuirá el tráfico IP que origina su máquina. Esto es especialmente importante si paga

por cada byte que circule por el cable. Tiene el valor añadido de evitarle el deber del mantenimiento; no tiene porqué actualizar un archivo `root.cache` vacío.

7. PUFs *Preguntas de Uso Frecuente (FAQ)*

En esta sección incluyo algunas de las preguntas más frecuentes realizadas relativas a DNS y este COMO. Y las respuestas :-). Por favor, lea esta sección antes de enviarme correo electrónico.

7.1 ¿Cómo uso DNS desde dentro de un cortafuegos (*firewall*)?

Unas cuantas pistas: 'forwarders', 'slave', y echar un vistazo a la literatura que hay al final de este COMO.

8.2 ¿Cómo hago que DNS rote entre las direcciones disponibles para un servicio, por ejemplo para `www.siempre.ocupado` para obtener balanceo de carga o similar?

Haga varios registros A para `www.siempre.ocupado` y use `bind 4.9.3` o posterior. `bind` hará una rotación tipo *round-robin* de las respuestas. Esto no funcionará con versiones anteriores de `bind`.

7.3 Quiero configurar DNS en una intranet (cerrada) ¿qué hago?

Elimine el archivo de caché y haga los archivos de zona. Eso también significa que nunca tendrá que actualizar el archivo de caché.

7.4 Mi sistema no tiene el programa `ndc`. ¿Qué hago?

El `bind` instalado en su sistema es viejo y de alguna forma obsoleto. Si la seguridad es importante para Vd.: actualice `bind` inmediatamente. Si no, de todas formas podrá vivir con ello. En lugar de ejecutar `ndc start` ejecute `named`; `ndc reload` será `named.reload` y `ndc restart` será `named.restart`. Esos programas probablemente estarán en `/usr/sbin`.

7.5 ¿Cómo configuro un servidor de nombres secundario?

Si el servidor primario tiene la dirección 127.0.0.1, ponga la siguiente línea en el archivo `named.boot` de su secundario:

```
secondary linux.bogus 127.0.0.1 sz/linux.bogus
```

7.6 Quiero que `bind` se ejecute cuando me desconecto de la red.

He recibido este mail de Ian Clark, ic@deakin.edu.au donde explica la forma de hacer esto:

``Ejecuto `named` en la máquina que hace *masquerading* aquí. Tengo dos archivos `root.cache`, uno llamado `root.cache.real` que contiene el servidor de nombres raíz real y el otro llamado `root.cache.falso` que contiene...

```
; root.cache.falso
; este archivo no contiene informacion
```

Cuando dejo de estar conectado copio el archivo `root.cache.falso` en `root.cache` y reinicio `named`.

Cuando me conecto copio `root.cache.real` en `root.cache` y reinicio `named`.

Esto se hace desde `ip-down` & `ip-up` respectivamente.

La primera vez que efectúo una consulta *off line* sobre un nombre de dominio del cual `named` no tiene detalles, éste pone una entrada como esta en `messages`...

Jan 28 20:10:11 hazchem named[10147]: No root nameserver for class IN

con la cual puedo convivir sin problemas.

Esto ciertamente parece funcionar. Puedo usar el servidor de nombres para máquinas locales mientras no estoy conectado sin el retraso con nombres de dominio externos, y cuando sí estoy conectado, funciona de forma normal con dominios externos."

7.7 ¿Dónde almacena su caché el servidor de nombres? ¿Hay alguna forma de controlar el tamaño del caché?

El caché se almacena en memoria completamente. No se escribe en disco en ningún momento. Cada vez que mata a `named` se pierde el caché. El caché no es controlable de ninguna forma, `named` lo maneja de acuerdo con unas reglas simples. No puede controlar ni el caché ni su tamaño de ninguna forma ni por ningún motivo. Si quiere, puede cambiar esto tocando los fuentes de `named`, lo cual no es recomendable.

7.8 ¿Salva `named` el caché entre reinicios? ¿Puedo guardarlo?

No, `named` no salva el caché cuando muere. Esto significa que el caché se debe reconstruir de nuevo cada vez que mate y reinicie `named`. No hay forma de hacer que `named` salve el caché en un archivo. Si quiere, puede cambiar esto tocando los fuentes de `named`, lo cual no es recomendable.

8. Cómo hacerse un gran *admin* DNS.

Documentación y herramientas.

Existe *Documentación Real*. En línea e impresa. Se requiere la lectura de esta documentación para seguir los pasos de pequeño a gran *admin* DNS. En

formato impreso el libro estándar es *DNS and BIND* de C. Liu y P. Albitz de *O'Reilly & Associates*, Sebastopol, CA, ISBN 0-937175-82-X. Lo leí, es excelente. Hay también una sección sobre DNS en *TCP/IP Network Administration*, de Craig Hunt de *O'Reilly...*, ISBN 0-937175-82-X. Otros libros necesarios para un Buenos Administradores DNS (o bueno para cualquier cosa de la materia) es *Zen and the Art of Motorcycle Maintenance* de Robert M. Prisig :-) Disponible con ISBN 0688052304 y otros.

Puede encontrar material en línea en <http://www.dns.net/dnsrd/>, <http://www.vix.com/isc/bind/>; una *PUF*, un manual de referencia (*BOG*; *Bind Operations Guide*) así como *papers* y definición de protocolos y diversos retoques o *hacks* de DNS (éstos y la mayoría, si no todas las referencias mencionadas arriba, están también contenidas en la distribución de *bind*). No he leído la mayoría, pero tampoco soy un gran admin DNS. Arnt Gulbrandsen, por otra parte ha leído el *BOG* y está extasiado con él :-). El grupo de noticias comp.protocols.tcp-ip.domains es sobre DNS. En suma, hay un gran número de RFCs sobre DNS, las más importantes son probablemente las siguientes:

- RFC 2052** A. Gulbrandsen, P. Vixie, A DNS RR for specifying the location of services (DNS SRV), Octubre de 1996
- RFC 1918** Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear, *Address Allocation for Private Internets*, 02/29/1996.
- RFC 1912** D. Barr, Common DNS Operational and Configuration Errors, 02/28/1996.
- RFC 1713** A. Romao, *Tools for DNS debugging*, 11/03/1994.
- RFC 1712** C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *DNS Encoding of Geographical Location*, 11/01/1994.
- RFC 1183** R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *New DNS RR Definitions*, 10/08/1990.
- RFC 1035** P. Mockapetris, Domain names - implementation and specification, 11/01/1987.
- RFC 1034** P. Mockapetris, *Domain names - concepts and facilities*, 11/01/1987.

RFC 1033 M. Lottor, *Domain administrators operations guide*, 11/01/1987.

RFC 1032 M. Stahl, *Domain administrators guide*, 11/01/1987.

RFC 974 C. Partridge, *Mail routing and the domain system*, 01/01/1986.

9. Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el INSFLUG se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (Preguntas de Uso Frecuente, las *FAQs*. :), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las últimas versiones de las traducciones: www.insflug.org. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

Cronología del DNS*

<http://www.nic.mx/evol/niana.html>

Por: Oscar Robles/NIC-México

El objetivo del presente artículo es mostrar la evolución que ha presentado el DNS, y a grandes rasgos, los últimos acontecimientos en las cuestiones políticas del mismo.

Resulta interesante la evolución que han presentado los nombres de dominio en Internet¹, ya que hace sólo dos años los gobiernos permanecían al margen de este aspecto. Hoy en día, el gobierno de los EEUU es uno de los más interesados en el tema, y por muchas razones el más discutido.

A continuación la historia.

En Internet, la comunicación entre los equipos y los humanos se facilita por el hecho de que los primeros tienen asignado un nombre, de esta forma, recordamos más fácil el nombre de una máquina ya que podemos asociar este a la organización o lugar en el que se encuentra, sin tener que memorizar la dirección de IP del equipo. Por ejemplo, pocos de nosotros sabemos que la máquina www.cnn.com tiene las direcciones de IP 207.25.71.22 (una de ellas).

Este concepto se conoce como Sistema de Nombres de Dominio, (**DNS** por sus siglas en inglés, Domain Name System), el cuál nació en la década de los 80's. Creado por Paul Mockapetris en colaboración con Jon Postel de la Universidad del Sur de California y posteriormente, Paul Vixie. Juntos desarrollaron lo que hasta ahora conocemos como el DNS (**BIND**, Berkeley Internet Name Domain), un sistema cliente/servidor, distribuido y jerárquico, características que se describen a detalle en los RFC² 1033, 1034 y 1035 y que son muy parecidas a un sistema de archivos de UNIX... pero distribuido.

Originalmente, el uso del **DNS** involucró solamente instituciones académicas, de investigación y por supuesto, la milicia de los EEUU. Eran los tiempos en que las universidades empezaban a realizar su conexión a las múltiples redes, entre ellas BitNet. Algo empezaba a trascender y era importante establecer un orden en cuanto a los equipos que ingresaban a la red.

Se crearon entonces los nombres de dominio genéricos de primer nivel (**gTLD**, generic Top-level Domain), .com, .net y .org, es decir, se habían creado estas tres clasificaciones con el fin de ubicar el tipo de entidades que buscaban tener presencia en Internet. Además de estos **gTLD** se empezó por delegar los sufijos nacionales (**nTLD**, national Top-level Domain) a los países que se fueran conectando a la red. De esta forma, a México se le asignó el .mx a finales de 1988 cuando el ITESM, Campus Monterrey se conecta de manera dedicada al Internet, este **nTLD** empieza a operar desde 1ro. de Febrero de 1989. Así cada país obtuvo su propio **nTLD**, incluso EEUU, el cual tiene el .us. También existen unos nombres de dominio especiales, **sTLD**, que son sólo para los EEUU: .mil, .edu y .gov³.

Las organizaciones que administran los **nTLD** por lo general son instituciones académicas, como es el caso del .mx y el ITESM, sin embargo el caso de los **gTLD** es diferente, estos originalmente fueron administrados por el Stanford Research Institute Network Information Center (**SRI-NIC**), de la Universidad de Stanford en Menlo Park, California, pero pronto cambiaría a InterNIC.

En 1992, la Fundación Nacional de Ciencias de los EEUU (NSF, National Science Foundation) quien administraba el backbone de Internet (en ese entonces NSFNET) decide licitar la operación del InterNIC y en 1993, a través de un convenio de cooperación, le otorga esta función a la empresa Network Solutions Inc.(NSI), posteriormente, esta empresa sería adquirida por el grupo Science Application International Corporation, (SAIC), originario de San Diego y que se distingue por sus multimillonarios contratos federales (Agencia de Seguridad Nacional, CIA, Marina de los EEUU)

Cuando NSI obtuvo el contrato, se estableció un apoyo de cuatro millones USD por parte de la NSF a NSI, para realizar la función del registro de los gTLD. No obstante, en 1994, el grupo SAIC compra esta empresa y su experiencia en contratos federales le ayuda a renegociar el contrato previo. De esta forma, logra que se empiece a cobrar \$50 USD anuales por cada nombre de dominio, estableciendo que el 30% de estas cuotas se irían a un fondo de infraestructura administrado por la NSF.

Aquí empezaron los problemas, aunque la operación de NSI fue buena, en mi opinión, muchos consideraban que el servicio debía mejorarse, tomando en cuenta las utilidades que esta empresa percibía por la operación de InterNIC. De hecho, este margen de utilidad le permitió a NSI empezar a cotizarse en la bolsa (NASDAQ: NSOL).

Para mediados de 1996, Jon Postel, el director del Internet Assigned Numbers Authority (IANA), y organismo administrador de las direcciones de IP y nombres de dominio, realizó una propuesta en la que contemplaba la creación de 150 nuevos nombres de dominios genéricos, gTLD, así como el .com, .net y .org. Esta propuesta pronto tuvo efectos importantes y finalizó en la formación de un grupo que se encargaría de discutir el re-diseño de los gTLD.

De esta forma, en Noviembre de 1996 nació el Internet-International Ad Hoc Committee (IAHC) impulsado por la Internet Society (ISOC), a los tres meses de haberse creado se generó el reporte final, en donde se planteaban las recomendaciones y requerimientos para un nuevo esquema de gTLD, este documento recibiría el nombre de Memorando de Entendimiento para los Nombres de Dominio genéricos de Nivel Superior.

El IAHC se disolvió en Mayo del 97 para dar paso al generic Top level Domain Memorandum of Understanding (gTLD-MoU), documento respaldado por organizaciones de todo el mundo, entre ellas la Organización Mundial de la Propiedad Industrial (WIPO), Union Internacional de Telecomunicaciones (ITU), Internet Society, MCI y por Latino América sólo NIC-México. En este documento se plasman los acuerdos alcanzados durante esos ocho meses de discusión y consenso, en los que por cierto no estuvo ningún representante del gobierno de país alguno.

Todo marchaba sobre ruedas, el gTLD-MoU contemplaba nuevos gTLD (.firm, .shop, .web, .arts, .rec, .info, .nom, una administración múltiple y distribuida de los gTLD, por ejemplo, que más de una organización pudiera registrar nombres de dominio bajo .com, la creación de un consejo central (CORE, Council of Registrars) formada por las organizaciones que fungirían como nuevos InterNICs, y dos cuerpos más de soporte al nuevo esquema, Policy Advisory Group (PAB) y el Policy Oversight Committee (POC). Así mismo, se contemplaba en esta propuesta un esquema que permitiera cambiar de registro, es decir, portabilidad de los nombres de dominio, esto aseguraba, según el gTLD-MoU, que todos los registros dieran un servicio de calidad.

Así, el nuevo esquema requería de un inversión para el complejo sistema que debería consolidar la información de los 89 registros aceptados. Para esto **CORE** estableció un contrato con Emergent Corp para el desarrollo del nuevo esquema distribuido de DNS (new DNS Shared Registry System). Todo estaba listo pues, para que en Marzo de 1998 empezaran a operar los 89 registros en todo el mundo, aceptando así solicitudes de dominio bajo los siete nuevos nombres de dominio genéricos.

El 30 de Enero, menos de dos meses antes de la fecha de inicio de operaciones del gTLD-MoU, el Gobierno de los Estados Unidos hace acto de presencia, en algo que parecía no apto para políticos. A través del Departamento de Comercio (DoC) publica un documento, conocido como Green Paper, en el cual, Ira Magaziner, asesor de Bill Clinton establece la postura de la Casa Blanca en materia de nombres de dominio. No pasaron muchas horas para que los principales actores de la industria enviaran sus comentarios al **DoC** expresando su desacuerdo.

En resumen, este documento desconocía la autoridad y el consenso del gTLD-MoU y por lo tanto de las organizaciones que lo representaban, a pesar de que el **CORE** ya estaba listo para iniciar operaciones. Su propuesta era esperar y planear mejor las cosas, como si veinte meses no hubieran sido suficientes en un medio ambiente tan dinámico como es Internet, y se concretaba a proponer una nueva organización que supliera al **IANA** en la coordinación de los gTLD y direcciones de IP para empezar funciones el 30 de Septiembre de 1998.

Este hecho provocó una cantidad impresionante de comentarios en contra del **Green Paper**, los cuales según el **DoC**, se tomarían en cuenta para generar una iniciativa global. Así, el 5 de Junio de 1998, el Gobierno de los EEUU, a través del **DoC** emitió un documento conocido como White Paper, en el cual, prácticamente se retractaban de algunos aspectos que habían planteado originalmente en el Green Paper.

Aunque no tan polémico como el anterior, este documento era de alguna forma los planteamientos finales del Gobierno de los EEUU para realizar la transición en la administración de Internet. A grandes rasgos, se limitaba a buscar una nueva organización central que supliera al **IANA**, de hecho se buscaba que fuera privatizada (sin fondos del gobierno) pero que fuera sin fines de lucro. Los aspectos en la generación de nuevos nombres de dominio genéricos, gTLD, se dejaban a consideración de esta nueva organización, conocida ahora como el Nuevo IANA (nIANA).

El tiempo cada vez era más corto, por lo que surgieron organizaciones que buscaban acelerar las decisiones necesarias para el lanzamiento del **nIANA**. Entre ellas, una organización con las siglas **GIAW**, que posteriormente fue **IFWP**, (International Forum for the White Paper). Fue esta organización quien se encargó de realizar la primer consulta para discutir puntos esenciales en el cumplimiento del White Paper, y le llamó la consulta de las Américas, la cual tuvo lugar en Reston VA, EEUU. A pesar del enfoque continental que se intentó darle, sólo hubo tres representantes de América Latina, **CABASE** de Argentina y **NIC-México**.

A esta reunión le siguieron otras convocadas por la Asociación de ISP en Europa (**EuroISPA**) y por la Comisión Europea en Bruselas Bélgica, teniendo como resultado un consenso más global, representativo del continente europeo.

Es un hecho que el actual staff del **IANA** se mantendrá de manera operativa en el **nIANA**, sin embargo esto no es materia de discusión. Lo que se pretende definir es la estructura

del **nIANA**. Se han establecido ya algunas reglas, en las cuales se plantean tres cuerpos que se encargarían de las direcciones de IP, los nombres de dominio de primer nivel **TLD** y los protocolos, respectivamente, además de un cuarto, en el que estarían representados los intereses de los primeros tres, así como de los principales grupos de interés (**stakeholders**) de Internet (usuarios, ISPs, etc).

La segunda reunión convocada por el **IFWP** fue sostenida en Ginebra, Suiza, en el marco del **INET'98**, una serie de eventos que realiza la **ISOC** anualmente. De todas las reuniones, esta ha sido sin duda la más representativa, con participantes latinos y africanos, inclusive. Se realizará una tercera reunión en Singapur, para escuchar las propuestas de Asia-Pacífico. Una cuarta reunión forma parte del calendario del **IFWP**, a desarrollarse en Buenos Aires, Argentina, en la cual se busca atraer el público latino que estuvo ausente en Reston, Virginia.

El resultado de estas reuniones de trabajo dará la pauta en el establecimientos de las reglas que aplicarán a Internet en los próximos años. Es imperativo que los stakeholders se involucren de manera activa en estos procesos y colaboren en la definición de lo que hoy día es su negocio.

1 Algunos autores utilizan esta palabra con un género: La internet, porque es la red; El Internet, porque es un servicio de valor agregado, prefiero utilizarlo sin género: En Internet.

2 RFC-Request For Comments, son una serie de documentos informativos públicos utilizados en Internet con la intención de convertirse en estándares o de reglamentar algún esquema de operación de redes, informática o sistemas.

3 En realidad es más compleja la organización de los nombres de dominio, para efectos prácticos basta con este esquema para comprender el artículo.

* Originalmente, una parte de este artículo apareció por primera vez en Mayo de 1998 en la revista electrónica VISION, del ITESM, Campus Monterrey.

Referencias e información adicional:

gTLD-MoU

DNS news

WIRED

NetNames

Internet World

TechWEB

PCWEEK

IFWP

IANA

Última modificación: August 05, 1998

[[Home](#)] [[Dominios](#)] [[IP's](#)] [[In-a](#)] [[Asn](#)] [[ISP](#)]
webmaster@nic.mx

NIC-Internet Costa Rica

Información de Dominios

<http://www.nic.cr/consulta-dns.html>

Top of Form 1

La consulta de un dominio es interactiva desde el Servidor del NIC a los Servidores que administran la Zona de Autoridad del dominio consultado. Por consiguiente, la respuesta a una consulta tiene una duración variable.

Bottom of Form 1

Dominios del Sector AC.cr de C.R.

SECTOR ACADEMICO

ac.cr: academico e investigacion.

ANC (Academia Nacional de las Ciencias)

anc.ac.cr. IN NS ns.anc.ac.cr.
ns.anc.ac.cr. IN A 163.178.20.2

UCR (Universidad de Costa Rica)

ucr.ac.cr. IN NS ns.ucr.ac.cr.
IN NS cariari.ucr.ac.cr.
ns.ucr.ac.cr. IN A 163.178.88.2
cariari.ucr.ac.cr. IN A 163.178.101.5

CIT/ITCR (Centro de Información)

cett.itcr.ac.cr. IN MX 10 zapote2.cit.itcr.ac.cr.
cit.itcr.ac.cr. IN NS zapote2.cit.itcr.ac.cr.
IN NS zapote.cit.itcr.ac.cr.
IN NS loro.cic.itcr.ac.cr.
zapote2.cit.itcr.ac.cr. IN A 163.178.132.35
zapote.cit.itcr.ac.cr. IN A 163.178.132.2

ITCR (Instituto Tecnológico de Costa Rica)

itcr.ac.cr. IN NS ns.itcr.ac.cr.
IN NS himalia.cic.itcr.ac.cr.
ns.itcr.ac.cr. IN A 196.40.16.2
himalia.cic.itcr.ac.cr. IN A 196.40.16.112

IC-ITCR (IC-ITCR)

ic-iter.ac.cr. IN NS ns.ic-iter.ac.cr.
IN NS ns2.ic-iter.ac.cr.
ns.ic-iter.ac.cr. IN A 196.40.30.178
ns2.ic-iter.ac.cr. IN A 196.40.30.179

UNED (Universidad Estatal a Distancia)

uned.ac.cr. IN NS arenal.uned.ac.cr.
arenal.uned.ac.cr. IN A 163.178.32.2

FOD (Fundacion Omar Dengo)

fod.ac.cr. IN NS rgdweb.fod.ac.cr.
IN NS rtepxy01.fod.ac.cr.
rgdweb.fod.ac.cr. IN A 196.40.30.40
rtepxy01.fod.ac.cr. IN A 163.178.200.2

INCAE (Instituto Centroamericano de Administracion de Empresas)

incae.ac.cr. IN NS fw.incae.ac.cr.
IN NS rip.psg.com.
rip.psg.com. IN A 147.28.0.39
fw.incae.ac.cr. IN A 196.40.23.90

INBIO (Instituto Nacional de Biodiversidad)

inbio.ac.cr. IN NS fulgora.inbio.ac.cr.
IN NS rip.psg.com.
fulgora.inbio.ac.cr. IN A 196.40.25.146

IICA (Instituto Interamericano de Cooperacion para la Agricultura)

iica.ac.cr. IN NS tlaloc.iica.ac.cr.
IN NS osiris.iica.ac.cr.
tlaloc.iica.ac.cr. IN A 163.178.36.151
osiris.iica.ac.cr. IN A 163.178.36.152

EARTH (Escuela de Agricultura de la Region Tropical Humeda)

earth.ac.cr. IN NS ns.earth.ac.cr.
IN NS pascal.ice.co.cr.
ns.earth.ac.cr. IN A 208.133.203.147

MEDVET (Medicina Veterinario de la UNA)

medvet.una.ac.cr. IN NS ns.medvet.una.ac.cr.
ns.medvet.una.ac.cr. IN A 163.178.148.2

UNA (Univeridad Nacional)

una.ac.cr. IN NS samara.una.ac.cr.
IN NS tambor.una.ac.cr.
samara.una.ac.cr. IN A 163.178.140.3
tambor.una.ac.cr. IN A 163.178.140.21

CATIE (Centro Agronomico Tropical de Investigacion y Ensenanza)

catie.ac.cr. IN NS computo.catie.ac.cr.
computo.catie.ac.cr. IN A 196.40.0.243

ULACIT (Universidad Latinoamericana de Ciencia y Tecnologia)

ulacit.ac.cr. IN NS servdoc.ulacit.ac.cr.
IN NS servadm.ulacit.ac.cr.
servdoc.ulacit.ac.cr. IN A 196.40.2.130
servadm.ulacit.ac.cr. IN A 196.40.2.138

OTS (Organizacion de Estudios Tropicales)

ots.ac.cr. IN NS andira.ots.ac.cr.
IN NS rip.psg.com.
andira.ots.ac.cr. IN A 207.1.115.98
rip.psg.com. IN A 147.28.0.39

ULATINA (Universidad Latina de Costa Rica)

ulatina.ac.cr. IN NS ns.ulatina.ac.cr.
IN NS ns.dxii.ulatina.ac.cr.
ns.ulatina.ac.cr. IN A 163.178.60.2
ns.dxii.ulatina.ac.cr. IN A 206.153.39.108

ECOUNCIL (Consejo de la Tierra)

ecouncil.ac.cr. IN NS ns.ecouncil.ac.cr.
ns.ecouncil.ac.cr. IN A 196.40.1.51

IMN (Instituto Meteorologico Nacional)

imn.ac.cr. IN NS meteo.imn.ac.cr.
meteo.imn.ac.cr. IN A 163.178.56.2

UICR (Universidad Interamericana de Costa Rica)

uicr.ac.cr. IN NS ns.uicr.ac.cr.
IN NS terraba.cc.uicr.ac.cr.
ns.uicr.ac.cr. IN A 206.153.35.242
terraba.cc.uicr.ac.cr. IN A 206.153.35.243

CEFOF (Centro de Formador de Formadores)

cefof.ac.cr. IN NS ns.cefof.ac.cr.
ns.cefof.ac.cr. IN A 163.178.62.2

UVERITAS (Universidad Veritas)

uveritas.ac.cr. IN NS ns.uveritas.ac.cr.
ns.uveritas.ac.cr. IN A 196.40.43.98

VERITAS (Universidad Veritas)

veritas.ac.cr. IN NS ns.uveritas.ac.cr.

ILAM (Instituto Latinoamericano de Museologia) DV (Kitcom)

ilam.ac.cr. IN NS ns.nameservers.net.
IN NS ns2.nameservers.net.
ns.nameservers.net. IN A 207.158.192.40
ns2.nameservers.net. IN A 209.41.31.13

DOMUS (Universidad Latina de C.R.)

domus.ac.cr. IN NS ns2.domus.ac.cr.
IN NS ns.ulatina.ac.cr.
ns2.domus.ac.cr. IN A 163.178.61.2

UIA (Universidad Internacional de las Americas)

uia.ac.cr. IN NS ns.uia.ac.cr.
ns.uia.ac.cr. IN A 163.178.76.2

CONARE (Consejo Nacional de Rectores)

conare.ac.cr. IN NS ns.conare.ac.cr.
ns.conare.ac.cr. IN A 163.178.80.2

UACA (Universidad Autonoma de Centro America)

uaca.ac.cr. IN NS ns.nameservers.net.
IN NS ns2.nameservers.net.

ICAP (Instituto Centroamericano de Administracion Publica) DV (Publinet)

icap.ac.cr. IN NS ns.publinet.co.cr.

IIHR (Instituto Interamericano de Derechos Humanos) DV (Axioma)

iihr.ac.cr. IN NS iidhweb.ac.cr.
iidhweb.ac.cr. IN A 206.153.35.11

ACGUANACASTE (Areas de Conservacion Guanacaste) DV (Publinet)

acguanacaste.ac.cr. IN NS ns.publinet.co.cr.

CSUCA (Consejo Superior)

csuca.ac.cr. IN NS puerta.csuca.ac.cr.
puerta.csuca.ac.cr. IN A 163.178.66.2

UGAM (Universidad Joaquin Garcia Monge) DV (Nexos)

ugam.ac.cr. IN NS nexint.nexos.co.cr.

UHIPANOAMERICANA (Universidad Hispanoamericana).

uhispanoamericana.ac.cr. IN NS ns.uhispanoamericana.ac.cr.
ns.uhispanoamericana.ac.cr. IN A 163.178.82.2

NEWHORIZONS (Aplica S.A. New Horizons Costa Rica) DV (La Nacion)

newhorizons.ac.cr. IN NS ns.newhorizons.ac.cr.
IN NS pino.newhorizons.ac.cr.
ns.newhorizons.ac.cr. IN A 196.40.9.82
pino.newhorizons.ac.cr. IN A 196.40.9.83

FUNDEPOS (Fundacion estudios de postgrado e investigacion en ciencias economicas)

fundepos.ac.cr. IN NS ns.fundepos.ac.cr.
ns.fundepos.ac.cr. IN A 206.153.35.226

UFIDELITAS (Universidad Fidelitas)

ufidelitas.ac.cr. IN NS ns.ufidelitas.ac.cr.
ns.ufidelitas.ac.cr. IN A 163.178.124.2

CIDH (Centro Internacional para el Desarrollo Humano) DV (Fundacion Acceso)

cidh.ac.cr. IN NS iguana.internexo.co.cr.
IN NS rip.psg.com.

IPJIMENEZ (Instituto Parauniversitario Jimenez) DV (intnet S.A.)

ipjimenez.ac.cr. IN NS ns.ipjimenez.ac.cr.
ns.ipjimenez.ac.cr. IN A 206.153.38.226

CIPET (Centro de Investigaci3n y Perfeccionamiento para la Educaci3n T3cnica)

cipet.ac.cr. IN NS cipetnt.cipet.ac.cr.
cipetnt.cipet.ac.cr. IN A 196.40.22.18

ABOGADOS (Colegio de Abogados de Costa Rica) DV (Goldnet)

abogados.ac.cr. IN NS abogados.ac.cr.
abogados.ac.cr. IN A 200.9.63.180

CIPET (Centro de Investigaci3n y Perfeccionamiento para la Educaci3n T3cnica)

cipet.ac.cr. IN NS cipetnt.cipet.ac.cr.
cipetnt.ac.cr. IN A 200.9.62.114

INA (Instituto Nacional de Aprendizaje)

ina.ac.cr. IN NS tequila.ina.ac.cr.
IN NS ron.ina.ac.cr.
tequila.ina.ac.cr. IN A 196.40.44.146
ron.ina.ac.cr. IN A 196.40.44.147

UCENTRAL (Fundacion UC) DV (Fundacion Acceso)

ucentral.ac.cr. IN NS iguana.internexo.co.cr.
IN NS rip.psg.com.

LEFRANCO (Liceo Franco Costarricense) DV (Internet de Centroamerica)

lefranco.ac.cr. IN NS ns1.edson.net.
IN NS ns2.edson.net.
ns1.edson.net. IN A 192.41.1.43
ns2.edson.net. IN A 192.41.2.43

ULASALLE (Universidad De La Salle)

ulasalle.ac.cr. IN NS ns.ulasalle.ac.cr.
ns.ulasalle.ac.cr. IN A 208.56.103.160
ns2.ulasalle.ac.cr. IN A 208.56.233.142

CIAPA (Centro de Investigacion y Adiestramiento Politico Administrativo) DV (AmericalatinA.net S.A.)

ciapa.ac.cr. IN NS fractal.icr.co.cr.
IN NS lorenz.icr.co.cr.
lorenz.icr.co.cr. IN A 206.153.32.133

USJ (Fundacion Universidad San Jose) DV (Fundacion Acceso)

usj.ac.cr. IN NS iguana.internexo.co.cr.
IN NS rip.psg.com.

UFEDERADA (Fundacion San Tomas de Aquino) DV (Interactive Creations)

ufederada.ac.cr. IN NS ns1.secure.net.
IN NS ns2.secure.net.

UCI (Universidad para la Cooperacion Int.)

uci.ac.cr. IN NS ns.siteprotect.com.
IN NS ns2.siteprotect.com.
ns.siteprotect.com. IN A 209.100.98.10
ns2.siteprotect.com. IN A 209.224.144.2
cihost.com. IN A 216.71.93.147

ALIANZAFR (Asoc. Alianza Cultural Franco-Costarricense) DV (Ticoweb)
alianzafr.ac.cr. IN NS ns.media3.net.
IN NS ns2.media3.net.
ns.media3.net. IN A 208.226.167.250
ns2.media3.net. IN A 206.67.48.250

ECUMENICA (Revista Ecumenica) DV (Funda. Acceso)
ecumenica.ac.cr. IN NS iguana.internexo.co.cr.

MADC (Fundacion Centro Nacional de la Cultura)
madc.ac.cr. IN NS ns.madc.ac.cr.
ns.madc.ac.cr. IN A 163.178.101.201

ALMAMATER (Universidad Alma Mater S.A)
almamater.ac.cr. IN NS ns.fundepos.ac.cr.

ULICORI (Instituto Graduado de Investigacion en Planificacisn y
; Desarrollo PLEURUS) DV (Tecmedia)
ulicori.ac.cr. IN NS ns1.expreso.co.cr.
IN NS ns2.expreso.co.cr.

UNICOR (Universidad Internacional de Costa Rica)
unicor.ac.cr. IN NS ns1.internet-dns.net.
IN NS ns2.internet-dns.net.
ns1.internet-dns.net. IN A 208.238.102.2
ns2.internet-dns.net. IN A 208.238.102.3

ECAG (Escuela Centroamericana de Ganaderia) DV (Cybernet de CR)
ecag.ac.cr. IN NS triton.cybermall.co.cr.

COVAO (Colegio Vocacional de Artes y Oficios)
covao.ac.cr. IN NS ns.covao.ac.cr.
ns.covao.ac.cr. IN A 208.133.203.34

UPAZ (Universidad de la Paz) DV (Consejo de la Tierra)
upaz.ac.cr. IN NS ns.upaz.ac.cr.
ns.upaz.ac.cr. IN A 196.40.1.51

MEP (Ministerio de Educacion Publica)
mep.ac.cr. IN NS ns.mep.ac.cr.
ns.mep.ac.cr. IN A 163.178.104.142

UNIDIS (Fundacion Costarricense de Diseno y Arquitectura) DV
; (Mundo Digital)
unidis.ac.cr. IN NS ns1.expreso.co.cr.
IN NS ns2.expreso.co.cr.

UCATOLICA (Fundacion Universidad Catolica de C.R.)
ucatolica.ac.cr. IN NS ns1.conexion.co.cr.
IN NS ns2.conexion.co.cr.
ITEA (Itea)

UNEM (Universidad Empresarial De Costa Rica)
unem.ac.cr. IN NS ns1.internet-dns.net.
IN NS ns2.internet-dns.net.

CUNA (Colegio Universitario de Alajuela)
cuna.ac.cr. IN NS roble.cuna.ac.cr.
roble.cuna.ac.cr. IN A 208.133.203.66

USAM (FUNDACION MANUEL ARAGON)
usam.ac.cr. IN NS ns.usam.ac.cr.
ns.usam.ac.cr. IN A 196.40.9.99

UCEM (UNIVERSIDAD CENTROAMERICANA DE CIENCIAS
EMPRESARIALES) DV (Mundo
; Digital)
ucem.ac.cr. IN NS ns1.expreso.co.cr.
IN NS ns2.expreso.co.cr.

UCA (Universidad de Cartago Florencio del Castillo) DV (Tools)
uca.ac.cr. IN NS ns.softcomca.com.
IN NS ns2.softcomca.com.

Total de Dominios AC.cr : 68

Cómo instalar un servidor de DNS

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100.

Cómo instalar un servidor de DNS

<http://hp.fciencias.unam.mx/~amem/dns/dns.html>

17 de abril, 1997

Este documento pretende servir como guía para la instalación de un servidor de nombres. Se asume un conocimiento básico del funcionamiento de una red y de un sistema Unix típico.

1. El software BIND

- 1.1 Dónde conseguir el software
- 1.2 Cómo instalarlo
- 1.3 Cómo ejecutarlo

2. El archivo de arranque

- 2.1 directory
- 2.2 cache
- 2.3 primary
- 2.4 secondary
- 2.5 forwarders
- 2.6 Ejemplos de archivos de arranque

3. Los archivos de zonas

- 3.1 SOA
- 3.2 A
- 3.3 CNAME
- 3.4 NS
- 3.5 PTR
- 3.6 MX
- 3.7 Ejemplos de archivos de zonas

4. Verificando la instalación

- 4.1 Los mensajes de named
- 4.2 Usando nslookup

1. El software BIND

En una red TCP/IP típica, el *servidor de nombres* es una máquina encargada de hacer la conversión entre direcciones nombres de máquinas y direcciones IP y viceversa. El software utilizado para este fin fue desarrollado en la universidad de Berkeley, California, y es llamado por esta razón BIND.

El software BIND incluye un demonio llamado *named*, así como utilerías que permiten controlar este programa, que es el encargado de realizar las traducciones de nombres a IP

1.1 Dónde conseguir el software

En algunas versiones de Unix, como SCO ODE 2.0 y superiores, el software ya viene incluido, sin embargo el código fuente de última versión de BIND puede obtenerse en <ftp.vix.com>. Este software puede compilarse en casi cualquier versión moderna de Unix, aunque es necesario leer cuidadosamente las instrucciones que acompañan al código fuente para hacer las adecuaciones necesarias a cada ambiente.

Para algunas versiones del sistema operativo Linux es posible conseguir *paquetes* precompilados que facilitan en gran manera la instalación del software. Para RedHat Linux, por ejemplo, es posible obtener el paquete `bind-4.9.4-3.i386.rpm` del sitio oficial de RedHat, <ftp.redhat.com>. Un buen sitio para obtener el paquete correspondiente a la versión de Slakware Linux es <sunsite.unc.edu>

1.2 Cómo instalarlo

El proceso de instalación varía de una versión de Unix a otra. En SCO, por ejemplo, es necesario utilizar el *custom runtime system* para verificar que la instalación de TCP/IP haya sido realizada adecuadamente. La instalación a partir de los fuentes debe hacerse siguiendo las instrucciones que acompañan al código. Es necesario disponer de un ambiente de desarrollo completo

(compilador, make, etc.), compilar los fuentes y poner el ejecutable `named` y programas auxiliares en un directorio como `/usr/sbin`.

La instalación en una máquina con Linux RedHat es muy sencilla. Basta utilizar el siguiente comando en directorio que contiene los paquetes de software RedHat (RPMS):

```
rpm -i bind-4.9.3-3.i386.rpm
```

Los números de versión del paquete, así como el especificador de arquitectura (i386) pueden variar: lo mejor es verificar el nombre exacto del paquete mediante el comando `ls`.

1.3 Cómo ejecutarlo

El sistema operativo debería arrancar por sí solo el demonio `named` al terminar su propia secuencia de arranque. En los sistemas UNIX BSD (Linux Slakware, por ejemplo), esto se logra incluyendo en el archivo `rc.local` la línea:

```
named &
```

Los sistemas que, por el contrario, siguen la norma System V para los archivos de arranque, contienen directorios especiales con *scripts* de shell que levantan los programas necesarios en cada *runlevel*. El caso de Linux RedHat, es el más sencillo: la misma instalación mediante el comando `rpm` agrega en el directorio correspondiente el *script* necesario para levantar el demonio.

2. El archivo de arranque

El primer archivo en la configuración de un servidor de nombres es el archivo de arranque, que es leído por el demonio `named` cuando arranca para saber a qué dominios va a servir y dónde encuentra las tablas de máquinas y direcciones IP, entre otras cosas. Este archivo es generalmente `/etc/named.boot`. Un ejemplo típico de este archivo es el siguiente:

; Archivo /etc/named.boot para master.cfemex.com

directory /var/named

| | dominio | archivo | respaldo |
|------------|----------------------|---------------|----------|
| cache | . | db.cache | |
| primary | cfemex.com | db.cfemex.com | |
| primary | 16.159.in-addr.arpa | db.159.16 | |
| primary | 0.0.127.in-addr.arpa | db.local | |
| forwarders | 159.16.10.24 | | |

Las líneas que conforman el archivo se describen a continuación.

2.1 directory

Indica el directorio donde named debe buscar los archivos que aparecen en las líneas siguientes. Si named no encuentra esta línea, buscará los archivos en el directorio /etc.

2.2 cache

Esta línea indica el archivo que contiene las direcciones de los servidores de nombres para dominios superiores. Es muy importante porque sin ella nuestro servidor de nombres no podrá resolver ningún nombre excepto aquellos para los cuales es autoritativo.

En una máquina conectada a internet este archivo debe contener las direcciones de las máquinas encargadas de resolver dominios superiores (.com o .edu, por ejemplo). Un archivo de caché correcto y actualizado puede obtenerse de <ftp.rs.internic.net>.

En una red interna que no tiene acceso a internet, o en una que se encuentra detrás de un *firewall*, este archivo debe estar presente pero puede estar vacío, ya que no hay necesidad de resolver nombres para máquinas de ningún dominio superior.

El archivo de caché para un subdominio que no tiene acceso a Internet, o sea, a servidores raíz, debería apuntar a su servidor de dominio superior. Aquí está una muestra del archivo de master.cfemex.com:

```
.          99999999 IN NS      INTERNET.CFEMEX.COM
INTERNET.CFEMEX.COM 99999999 A      159.16.10.24
```

2.3 primary

Indica un dominio para el cual este servidor de nombres es autoritativo. Para convertir nombres en direcciones IP la sintaxis es sencilla; basta poner el nombre del dominio completo y el archivo que contiene los nombres de las máquinas con sus direcciones IP. Cuando se trata de convertir direcciones IP a nombres, la sintaxis puede parecer un poco extraña. La pieza de información siguiente a la palabra *primary* no es otra cosa que el segmento de direcciones IP para el cual esta máquina es autoritativa, sólo que con los números invertidos y las palabras *in-addr.arpa* añadidas después de un punto. Al final queda, por supuesto, el archivo que contiene el mapeo de direcciones a nombres. Por ejemplo: el servidor de nombres cuyo archivo de arranque aparece arriba resuelve direcciones para todas las máquinas bajo el dominio cfemex.com. Resulta que estas máquinas tienen IPs de la forma 159.16.x.x, por lo que la línea para la resolución reversa contiene el texto: 16.159.in-addr.arpa.

Una línea *primary* que por lo general está presente es la que resuelve la reversa para las IPs 127.0.0, que es la última línea *primary* mostrada en el archivo de ejemplo, más arriba.

2.4 secondary

Al igual que *primary*, esta línea indica un dominio para el cual somos autoritativos, pero del que no obtenemos la información de primera mano a través de un archivo, sino a través de otro servidor de nombres. Esta línea sirve para implementar redundancia: dos servidores de distintos dominios pueden

declararse secundarios mutuamente y, si alguno de los dos se cae, el otro podrá responder por ambos hasta que el servicio se restablezca. La pieza de información que sigue a la palabra *secondary* es la dirección del servidor de nombres del otro dominio, y a continuación tenemos un archivo donde *named* depositará la información que obtenga para la eventualidad de que el otro servidor se caiga, como ya dijimos.

2.5 forwarders

Esta línea sirve para indicar servidores de niveles superiores al nuestro, con el fin de redirigir las peticiones de información que el servidor de nombres no puede resolver. En este ejemplo ponemos la dirección del servidor de nombres externo de CFE porque suponemos que es el más indicado para resolver peticiones fuera de nuestra red.

2.6 Ejemplos de archivos de arranque

Para un zona subordinada

Supongamos que queremos dar de alta un servidor de nombres para una subzona de *cfemex.com*, por ejemplo *norte.cfemex.com*. El archivo de arranque debería ser como el siguiente:

; Archivo */etc/named.boot* para la zona *norte.cfemex.com*

```
directory /var/named
cache      .                db.cache
primary    norte.cfemex.com  db.norte.cfemex.com
primary    10.16.159.in-addr.arpa db.159.16.10
primary    0.0.127.in-addr.arpa db.local
forwarders 159.16.10.8
```

Nótese el uso de la línea *forwarders*: el servidor de nombres para la zona norte.cfemex.com necesita de un servidor de orden superior que le resuelva nombres que no estén en su dominio. Necesita también de un archivo de caché que contenga un servidor de orden superior:

```
.          99999999 IN NS    MASTER.CFEMEX.COM
MASTER.CFEMEX.COM  99999999 A    159.16.10.8
```

En este caso, el servidor de orden superior apunta a master.cfemex.com, que es el indicado para resolver nombres que quedan fuera de la jurisdicción de la zona norte.

Para un servidor que sólo es secundario

Un servidor secundario sólo utiliza líneas *secondary*, no se declara primario para ningún dominio (a excepción de la reversa de 127.0.0). Es el ejemplo típico de un servidor de respaldo, que se vuelve esencial cuando el servidor primario falla:

```
;
; Archivo de arranque para un servidor secundario
;
directory          /var/named
;
; Tipo      Dominio      servidor primario  respaldo
;
cache           .          db.cache
secondary  cfemex.com      master.cfemex.com  db.cfemex.com
primary      0.0.127.in-addr.arpa      db.local
```

La línea *secondary* indica el dominio para el cual se declara autoritativo, el servidor primario de donde recibe su información, y un archivo de respaldo, donde named guarda la información del dominio a modo de respaldo, para el caso de que el servidor primario falle.

Para un servidor *sólo caché*

Un servidor que sólo hace caché es un servidor que no es primario para ningún dominio, sólo redirige las peticiones de información hacia otro servidor y guarda las respuestas. Un servidor de nombres de este tipo es útil porque disminuye el tráfico de peticiones de resolución de nombres en la red. Un archivo de arranque típico para una máquina con acceso completo a Internet, es decir, que puede ver los servidores raíz, sería el siguiente:

```
Archivo de arranque para un servidor de sólo caché
directory /var/named
cache      .                db.cache
primary   0.0.127.in-addr.arpa  named.local
```

En caso de que el servidor de nombres no tenga acceso completo a Internet, el archivo deberá contener una línea *forwarders* y necesitará un archivo de caché similar al de la sección previa.

3. Los archivos de zonas

Como vimos antes, cada línea *primary* en el archivo `named.boot` asocia un dominio con un archivo dentro del directorio especificado por la línea *primary*. A estos archivos los llamamos archivos de zonas, y a su dominio asociado le llamamos origen. Cada nombre de dominio o de máquina que aparezca en estos archivos es considerado relativo a este origen a menos que termine con un punto. Esta regla no debe ser tomada a la ligera: todos los nombres en un archivo de zona son expandidos agregándoles el origen a menos que terminen en un punto, y esto nos puede desconcertar en ocasiones. Para hacer referencia al origen en sí es necesario usar el símbolo ``@''.

Los archivos de zonas están hechos de lo que en inglés se conoce como *resource records* o, simplemente, RRs, y tienen un tipo asociado. Por ejemplo, un registro para asociar un nombre de máquina con una dirección IP tiene el tipo

A, y uno para asociar un nombre corto o *alias* a una máquina a la que ya se le asoció un IP es de tipo *CNAME*. En general, los registros o RRs tienen la forma siguiente:

[dominio] [ttl] [clase] tipo datos

Cada campo del registro se separa mediante espacios o tabuladores. Los campos entre paréntesis cuadrados son opcionales, como se verá. Un campo puede ocupar más de una línea siempre y cuando aparezca un paréntesis antes del primer salto de línea y otro paréntesis cierre después del último campo del registro en la última línea. Un ejemplo típico de esto es el registro *SOA*, que veremos más adelante. Al igual que en el archivo de arranque, el *named.boot*, es posible introducir comentarios en un archivo de zona mediante un punto y coma: todo lo que aparece después de este signo es ignorado.

dominio

Este es el nombre del dominio (o de máquina, que para el caso es lo mismo), al cual se aplica el registro. Recordemos que sino termina en un punto se le considera relativo al origen del archivo. Este campo es opcional; si no aparece, el registro se aplica al RR anterior.

ttl

Del inglés *time to live*. Es posible asignarle a cada registro un tiempo de vida en segundos, que sirve para que cualquiera que utilice este registro descarte su información al término de este periodo.

Este campo es importante, porque si su valor es demasiado bajo, los clientes del servidor de nombres le harán solicitudes frecuentemente por el mismo dato, lo que afectará el rendimiento del sistema. Por el contrario, si este valor es muy alto es posible que los clientes no vean los cambios que puedan realizarse en la red (máquinas que cambian de IP, etc.). El valor típico de este campo es grande, especialmente si hablamos de redes de área local, donde la topología cambia lentamente.

Si este campo no parece en el registro, como es usual, el tiempo de vida es el que se menciona en un campo especial del registro *SOA* (*minimum*),

al principio del archivo de zona. Aquellos registros cuya información sea susceptible de cambio frecuente deberían tener un ttl explícito.

clase

Este campo tiene generalmente el valor */N*. Si no aparece este campo en el registro, se asume el del registro anterior.

tipo

Este campo indica el tipo del registro. Los tipos más comunes son *A*, *SOA*, *PTR* y *NS*. Los tipos se describen más abajo.

datos

El formato de estos datos esta relacionado con el campo anterior, el tipo.

Los tipos de registro más usados son los siguientes:

3.1 SOA

Este registro debe aparecer en cada archivo mencionado en una línea *primary*, e indica que los registros que siguen contienen información autoritativa para el dominio. *SOA* significa *Start of authority*. Este es un registro típico:

Archivo */var/named/db.cfemex.com*

```
@      IN SOA  master.cfemex.com. root.master.cfemex.com. (
      96112500      ; Serial
      10800        ; Refresh 3 horas
      3600         ; Retry 1 hora
      3600000      ; Expire 1000 hora
      86400       ; Minimum 24 horas
      )
      IN NS  master.cfemex.com.
```

Los datos asociados con un registro *SOA* son los siguientes:

origin

Es el nombre canónico del servidor de nombres primario para este dominio, y generalmente se da como absoluto, es decir, con un punto al final.

contact

Es el nombre de la persona responsable para este dominio. Es parecido a una dirección de correo electrónico normal, a excepción de que la arroba se reemplaza con un punto. También termina con un punto.

serial

Es un número que indica la versión del archivo de zona, y debe ser incrementado cada vez que el archivo se modifique. Es importante porque los servidores secundarios solicitan el registro *SOA* en ciertos intervalos (ver *refresh*, más abajo), para verificar el serial. Si éste ha cambiado, entonces transfieren el archivo completo para actualizarse.

Una práctica muy común es utilizar la fecha en el formato *ddmmaa* y agregarle dos dígitos más para los cambios que se hacen al archivo en el mismo día. De tal manera, un serial típico podría ser ``96112510": el décimo cambio realizado al archivo en el día 25 de noviembre de 1996.

refresh

Es el intervalo, en segundos, para las revisiones que hacen los servidores secundarios del registro *SOA*, con el fin de verificar si la información del dominio ha cambiado. El valor típico es de una hora (3600).

retry

Es el tiempo, en segundos, que un servidor secundario debe esperar para reintentar una conexión por *refresh* que ha fallado. El valor recomendado es de 10 minutos, o sea 600.

expire

Si un servidor secundario no ha podido comunicarse con su servidor primario para verificar que no haya habido cambios a la zona (mediante su registro *SOA*), descartará la información que tiene después de este periodo dado en segundos. El valor típico es de 42 días, o sea 3600000.

minimum

Este es el número de segundos empleado en los registros del archivo que no especifican su campo *ttl* (*time to live*).

3.2 A

Este registro sirve para asociar un nombre de máquina con una dirección IP. El único dato para este tipo de registro es la dirección IP en su forma estándar: *xxx.xxx.xxx.xxx*. Debe haber sólo un registro *A* por cada dirección IP en el archivo, aunque es posible asignarle a una máquina más de una dirección mediante varios registros *A*. Por ejemplo, asignémos a *pentium* las siguientes direcciones:

```
pentium IN  A   159.16.16.148
          IN  A   159.16.16.150
```

3.3 CNAME

Este registro sirve para asignarle un nombre alternativo o *alias* a una máquina. Supongamos que nuestra máquina *pentium* cumple las funciones de servidor de web y de noticias de nuestra red interna, *cfemex.com*. Una manera de hacer esto explícito a los usuarios de la red es usar los siguientes *alias*:

```
news  IN  CNAME pentium
www   IN  CNAME pentium
```

De esta manera tenemos, aunque sólo en apariencia, dos máquinas diferentes: news.cfemex.com y www.cfemex.com.

3.4 NS

Mediante un registro *NS* es posible designar un servidor que deberá responder para todas las peticiones que involucren un determinado subdominio. Esto es importante porque permite delegar la asignación de nombres y facilita el manejo de dominios complejos. El siguiente ejemplo designa al servidor lapem1.lapem.cfemex.com como servidor de nombres para el área lapem:

```
lapem IN NS lapem1.lapem
```

Designar un servidor de nombres, sin embargo, no basta. Necesitamos definir en alguna parte del archivo la dirección de este servidor, mediante un registro *A*, por supuesto. A este registro se le llama en inglés *glue record*.

3.5 PTR

Un registro *PTR* se utiliza para relacionar una dirección IP con un nombre de máquina, exactamente al revés que un registro tipo *A*. Estos registros aparecen en los archivos de zonas para la resolución reversa, los que en named.boot aparecen en una línea *primary* con el dominio *IN-ADDR.ARPA*.

```
82.10 IN PTR redes.cfemex.com.
```

Nótese que en cada registro sólo aparece una fracción de la dirección IP: la dirección se completa porque, como dijimos más arriba, a cada nombre que no termina en un punto se le agrega el origen.

Los nombres de máquinas aparecen siempre en los registros *PTR* en su forma canónica, es decir, con el dominio completo. El punto es necesario porque de no aparecer se le agregaría erróneamente el origen.

3.6 MX

Los registros *MX* sirven para anunciar a los programas de intercambio de correo (Sendmail, por ejemplo), una máquina que se encarga de administrar el correo de un determinado dominio. Un ejemplo típico es el siguiente:

```
@ IN MX 5 mailer
```

Esta línea declara a la máquina mailer como administradora de correo para el dominio origen. A diferencia de registros como el *A* o el *PTR*, el registro *MX* utiliza un campo numérico adicional que sirve para marcar la *preferencia* de la máquina. Un programa que quiera enviar correo a este dominio va a tratar de establecer contacto con todas las máquinas que tengan un registro *MX* asociado hasta que lo logre, pero va a tratar primero con el que tenga el número de preferencia más bajo.

3.7 Ejemplos de archivos de zonas

Para un subdominio

Este es un archivo típico de zona, y corresponde a lv.cfemex.com:

Zona 'lv.cfemex.com'

```
@ IN SOA sistlv11.lv.cfemex.com. root.sistlv11.lv.cfemex.com. (
    97012900
    43200
    3600
    360000
    360000
)
IN NS sistlv11.lv.cfemex.com.
IN NS systlv17.lv.cfemex.com.
IN NS master.cfemex.com.
```

```

sistlv10  IN  A  192.168.10.10
          IN  A  159.16.70.65
          IN  MX  5 sistlv10.lv.cfemex.com.
sistlv3   IN  MX  5 sistlv3.lv.cfemex.com.
          IN  A  192.168.10.3
sistlv11  IN  A  192.168.10.11
          IN  A  159.16.70.97
          IN  MX  2 sistlv11.lv.cfemex.com.

```

Para el mapeo inverso (de IPs a nombres)

A continuación mostramos un archivo para la resolución inversa, es decir, de direcciones IP a nombres:

Mapeo inverso de direcciones para 70.16.159.in-addr.arpa.

```

@          IN  SOA  sistlv11.lv.cfemex.com. root.sistlv11.lv.cfemex.com. (
          96052400
          43200
          3600
          360000
          360000
          )
          IN  NS   sistlv11.lv.cfemex.com.
          IN  NS   systlv17.lv.cfemex.com.
101        IN  PTR  h2020-3.lv.cfemex.com.
115        IN  PTR  clvpr1.lv.cfemex.com.
96         IN  PTR  vaxlv-net.
          IN  A    255.255.255.240
97         IN  PTR  sistlv11.lv.cfemex.com.
98         IN  PTR  vaxlv1.lv.cfemex.com.
146        IN  PTR  laser1in.lv.cfemex.com.
99         IN  PTR  vaxlv2.lv.cfemex.com.

```

Para el dominio 0.0.127.in-addr.arpa

Este archivo sirve únicamente para convertir la dirección 127.0.0.1 en localhost.

Mapeo inverso de direcciones para el dominio 0.0.127.in-addr.arpa.

```
@           IN      SOA    master.cfemex.com. root.master.cfemex.com. (
              97012900
              43200
              3600
              360000
              360000
            )
           IN      NS     master.cfemex.com.
1          IN      PTR    localhost.
```

4. Verificando la instalación

Casi cualquier problema relacionado con `named` que tenga su origen en la información proporcionada a través de los archivos de configuración puede ser diagnosticado a través de las utilerías de BIND y del registro que hace el mismo `named` en el `system logger`. A continuación veremos cómo lidiar con diversos problemas de configuración utilizando estas opciones.

4.1 Los mensajes de `named`

Todas las versiones de Unix cuentan con un registro de eventos o *system logger* que permite a las diversas aplicaciones que corren en el sistema registrar cualquier información que consideren pertinente en archivos especiales destinados a su revisión por parte del administrador del sistema. En SCO, Estos

archivos se encuentran localizados en el directorio `/usr/adm/`, en Linux en `/var/log` y tienen nombres como `messages` o `syslog`. El demonio `named` registra su arranque, por ejemplo, mediante una línea similar a la siguiente:

```
Dec 7 10:53:25 master named[29338]: starting. named 4.9.3-P1
```

Las lecturas a las bases de datos indicadas en el archivo de arranque (`named.boot`) mediante una línea `primary` son registradas de la siguiente manera:

```
Dec 7 10:53:25 master named[29338]: primary zone "cfemex.com" loaded  
(serial 97010201)
```

Las consultas a otros servidores de nombres para cargar zonas secundarias quedan registradas así:

```
Dec 7 10:53:25 master named[29338]: secondary zone "cfe.gob.mx" loaded  
(serial 96120300)
```

Finalmente, `named` marca el final de su arranque y su disposición a responder peticiones con una línea como la que sigue:

```
Dec 7 10:53:26 master named[29339]: Ready to answer queries.
```

Muchas anomalías pueden ser identificadas con sólo mirar los registros de `named`, entre ellas:

- Errores en la información contenida en los archivos de datos indicados en líneas `primary`. Generalmente se deben a una sintaxis incorrecta en una línea de asignación de IP. Es muy común también el error de asignar un alias (mediante *CNAME*) a una máquina cuyo IP no ha sido definido. Otro error común es olvidar que, a diferencia de otros tipos de registro, el registro *MX* utiliza un campo adicional, numérico, para indicar la preferencia de la máquina administradora de correo.
- Errores en la transferencia de información de un servidor primario a un servidor secundario. Estos errores pueden deberse a que el servidor primario no contestó la petición del servidor secundario porque estaba caído o porque simplemente no estaba corriendo el software BIND, o sea `named`. También es posible que la información recibida del servidor primario contuviera errores.
- Un archivo de caché incorrecto o faltante. Este error es generalmente fácil de resolver. Basta poner en el directorio referenciado en la línea *directory* del archivo `named.boot` un archivo de caché correcto.

Es posible recibir más información acerca de la operación del demonio `named` enviándole señales a través del comando `kill` de Unix, que se utiliza de la siguiente manera:

kill -s señal pid

donde *pid* es el número de proceso del demonio *named* actualmente corriendo, mismo que puede ser consultado mediante un comando como `ps -ax`. Las señales que se le pueden enviar a *named*, varias muy útiles en el diagnóstico de errores, son las siguientes:

SIGHUP

Causa que el servidor relea el archivo *named.boot* y recargue las bases de datos.

SIGINT

Provoca un vaciado de los datos cargados por *named* en el archivo */var/tmp/named_dump.db*.

SIGIOT

Provoca el vaciado de estadísticas en el archivo */var/tmp/named.stats*.

SIGSYS

Provoca el vaciado de información destinada al análisis del desempeño de *named* en *raw{/var/tmp}* (llamadas al sistema como *fork*, *chdir* o *exit*).

SIGTERM

Provoca el vaciado de las bases de datos primarias y secundarios, esto con el fin de guardar la información modificada cuando el demonio termina su operación.

SIGUSR1

Activa el modo de depuración. Cada envío de esta señal incrementa el nivel de depuración de *named*.

SIGUSR2

Desactiva completamente el modo de depuración.

SIGWINCH

Activa o desactiva el registro en el system logger de todas las consultas a `named`.

Información más detallada acerca del uso de señales para controlar a `named` puede encontrarse en la hoja de manual del programa (`man named`).

El paquete de instalación de BIND para Linux RedHat contiene un utilería interesante porque envuelve el uso de estas señales y proporciona al usuario una interfaz más sencilla. Esta utilería se llama `ndc`, por *named control*, y se utiliza de la siguiente manera:

```
ndc (status|dumpdb|reload|stats|trace|notrace|querylog|start|stop|restart)
```

Como puede observarse, las opciones que se le pueden indicar a `ndc` corresponden a las señales descritas con anterioridad. En Linux RedHat, `ndc` cuenta con una página de manual propia, al igual que `named`.

4.2 Usando `nslookup`

Otra utilería sumamente útil que se utiliza comunmente para verificar la instalación de un servidor de nombres es `nslookup`. Esta herramienta puede utilizarse interactivamente o como un comando cualquiera. Esto último se hace de la siguiente manera:

```
nslookup maquina
```

`nslookup` cuestionará al servidor de nombres especificado en el archivo `/etc/resolv.conf` acerca de la máquina especificada y devolverá su dirección IP.

En modo interactivo, `nslookup` puede hacer mucho más que sólo encontrar direcciones IP: puede preguntarle al servidor de nombres por cualquier clase de registros (no sólo A) e incluso puede mostrar la información referente a una zona entera.

Para entrar en el modo interactivo sólo es necesario teclear nslookup. El programa contestará con un signo de menor que ">" indicando que está listo para ejecutar comandos. Es posible entonces indicarle cualquier nombre de dominio y nslookup buscará por registros de tipo A. Para cambiar el tipo de registro que queremos encontrar es posible indicar set type=tipo, donde tipo puede ser cualquiera de los que ya mencionamos en secciones anteriores o incluso any, que indica cualquier tipo de registro.

La siguiente sesión, que se incluye como ejemplo, muestra cómo es posible encontrar información no sólo acerca de máquinas (direcciones IP), sino acerca de dominios (cuáles son sus servidores de nombres o de correo).

```
$ nslookup
```

```
Default Server: master.cfemex.com
```

```
Address: 159.16.10.8
```

```
> sunsite.unc.edu
```

```
Server: master.cfemex.com
```

```
Address: 159.16.10.8
```

```
Non-authoritative answer:
```

```
Name: sunsite.unc.edu
```

```
Address: 152.2.254.81
```

```
> unc.edu
```

```
Server: master.cfemex.com
```

```
Address: 159.16.10.8
```

```
*** No address (A) records available for unc.edu
```

```
> set type=SOA
```

```
> unc.edu
```

```
Server: master.cfemex.com
```

```
Address: 159.16.10.8
```

unc.edu

origin = ns.unc.edu

mail addr = host-reg.ns.unc.edu

serial = 970107010

refresh = 14400 (4 hours)

retry = 3600 (1 hour)

expire = 1209600 (14 days)

minimum ttl = 86400 (1 day)

unc.edu nameserver = ns.unc.edu

unc.edu nameserver = ns2.unc.edu

unc.edu nameserver = ncnoc.ncren.net

ns.unc.edu internet address = 152.2.21.1

ns2.unc.edu internet address = 152.2.253.100

ncnoc.ncren.net internet address = 192.101.21.1

ncnoc.ncren.net internet address = 128.109.193.1

> exit

\$

REQUISITOS DE CONFIGURACIÓN DE DNS PARA EL REGISTRO Y DELEGACIÓN DE UN DOMINIO DE SEGUNDO NIVEL BAJO "ES"

[HTTP://WWW.NIC.ES/DNSCONFIG/](http://www.nic.es/dnsconfig/)

La información relativa a los Requisitos de Configuración de DNS para el Registro y Delegación de un Dominio de Segundo Nivel bajo "es" que a continuación se presenta, corresponde a una versión en formato html del documento original de referencia que puede obtenerse vía ftp en: URL: <ftp://ftp.nic.es/docs/es-dom-dnsconfig.txt>

- 1.-Introducción.
- 2.-Requisitos de configuración de zonas de DNS de segundo nivel bajo "es".
- 3.-Lecturas Recomendadas.

1.- INTRODUCCIÓN

Antes de enviar un Formulario de Solicitud Electrónica (FSE) al ES-NIC para solicitar la delegación o cambios en la delegación de una zona de DNS de segundo nivel bajo "es" (ver Normas y Procedimientos para el Registro de un Nombre de Dominio de DNS bajo "es" en <ftp://ftp.nic.es/docs/es-dom-normas.txt>) los servidores de DNS especificados deben estar operativos, accesibles en Internet, dando respuestas autorizadas para el dominio en cuestión y correctamente configurados de acuerdo con las normas técnicas del DNS en general y de las recogidas en este documento en particular. En caso contrario se producirán retrasos en el registro, puesto que la solicitud tendrá que ser reenviada una vez corregidos los problemas que hayan sido detectados en los chequeos efectuados por el ES-NIC.

Toda zona de DNS de segundo nivel delegada bajo "es" habrá de ser mantenida y gestionada de forma técnicamente competente, esto implica que los servidores de DNS en donde este delegada deben permanecer (al igual que lo estaban en el momento de la delegación) operativos, accesibles en Internet, dando respuestas autorizadas para el dominio en cuestión y correctamente configurados de acuerdo con las normas técnicas del DNS en general y de las recogidas en este apartado en particular.

Cualquier cambio que afecte a la delegación de un dominio de segundo nivel bajo "es" (alta, baja o modificación de nombre o dirección IP de servidores de DNS para el dominio) debe ser prevista y planificada con antelación suficiente y solicitada al ES-NIC mediante el procedimiento establecido al efecto.

En caso de incompetencia o negligencia técnica, un dominio de DNS registrado y delegado bajo "es", puede ser dado de baja de forma temporal o definitiva.

<http://www.nic.es/normas/index.html>

2.- Requisitos de Configuración de zonas de DNS de segundo nivel bajo "es"

2.1.- Para la delegación de un dominio de DNS de segundo nivel bajo "es" es necesario disponer al menos de dos servidores autorizados de DNS para el dominio en cuestión: uno configurado como servidor primario y como mínimo, otro configurado como servidor secundario (aparte del que opcionalmente ofrece el ES-NIC). Para una mejor redundancia, se recomienda que los servidores autorizados para el dominio sean más de dos (hasta un máximo de siete) y estén físicamente en redes y localizaciones distintas (por ejemplo, el primario en la organización titular del dominio y un secundario en el proveedor que le da acceso a Internet).

2.2.- Los nombres de los servidores autorizados especificados en la solicitud al ES-NIC deben ser los nombres canónicos de dichos equipos (nunca un alias) y deben coincidir con los que aparezcan apuntados por registros "NS" en el servidor primario de la zona en cuestión. En el caso del servidor primario, el nombre debe también coincidir con el que en el propio servidor de DNS aparezca como origen en el SOA de la zona.

2.3.- Todos los servidores de DNS autorizados de toda zona de segundo nivel bajo "es" deben estar accesibles desde cualquier punto de Internet en general, y desde el servidor primario de "es" en particular (al menos por medio de ICMP y TCP y UDP al puerto 53).

2.4.- Todos los los servidores de DNS autorizados de toda zona de segundo nivel bajo "es" deben estar dando respuestas autorizadas ("authoritative answer") y consistentes para la zona en cuestión.

2.5.- Toda la información contenida dentro de un dominio bajo "es" (a cualquier nivel) deber ser correcta y contar con las autorizaciones que sean precisas. En concreto, esta terminantemente prohibido poner registros de cualquier tipo ("NS", "A", "MX", "CNAME", etc.), en o bajo cualquier zona de DNS de segundo nivel bajo "es", apuntando a equipos externos a la propia organización titular del dominio en cuestión sin contar con autorización expresa para ello. Esta acción puede dar lugar a la baja del dominio ofensor.

2.6.- Esta terminantemente prohibido efectuar "lame delegations" en o bajo cualquier zona de DNS de segundo nivel bajo "es". Una "lame delegation" consiste en poner un registro "NS" para un dominio o subdominio apuntando a un equipo que no esta configurado como servidor autorizado para dicha zona. Esta acción es más grave aun si la máquina a la que se está apuntando de forma errónea es de un tercero externo a la propia organización titular del dominio y puede dar lugar a la baja de este.

2.7.- El servidor primario de cualquier zona de segundo o inferior nivel bajo "es" debe poner "glue records" (es decir, registros de tipo "A" para servidores autorizados de la zona) únicamente para aquellos servidores que se encuentren bajo el propio dominio asociado a la zona y nunca para aquellos que sean externos al mismo. Esta acción puede dar lugar a la baja de un dominio.

2.8.- Esta terminantemente prohibido que una organización configure un servidor de DNS (aunque sean de "solo cache") como "forwarder" apuntando a un servidor externo del que no se tenga permiso explícito para hacerlo.

2.9.- Los clientes de DNS de una organización deben apuntar únicamente a servidores de DNS internos (o externos de los que se cuente con permiso explícito).

2.10.- El fichero de "cache" o de los servidores de la raíz (normalmente denominado "named.root", "named.cache" o "named.ca") de todo servidor de DNS debe estar siempre actualizado con la ultima versión del mismo que puede ser encontrada en: <ftp://ftp.nic.es/dns/named.root>

2.11.- Con el fin de servir de ayuda a la hora de detectar y solucionar problemas e inconsistencias entre el servidor primario y secundarios el formato del número serie del SOA recomendado para cualquier zona de DNS por debajo de "es" (a cualquier nivel) es el siguiente:

AAAAMMDDXX

es decir, 10 dígitos, donde:

AAAA = año de la última modificación de datos en la zona

MM = mes de la última modificación de datos en la zona

DD = día del mes de la última modificación de datos en la zona

XX = número de modificación de datos en la zona en el día

Por ejemplo, el número serie 1996080103 en el SOA de una zona se correspondería con la tercera modificación de la misma del día 1 de agosto de 1996.

2.12.- En el caso de que un servidor del ES-NIC este haciendo de servidor secundario opcional para la zona de segundo nivel bajo "es" en cuestión, el formato del número serie mencionado en el apartado anterior no es "recomendado" sino obligatorio.

2.13.- Con el fin de optimizar al máximo la relación "tráfico de DNS / rapidez de propagación de cambios en DNS", los valores normales recomendados de los parámetros temporales del SOA para zonas de DNS de segundo nivel bajo "es" son los siguientes:

Refresco ("refresh") = 86400 (1 día)

Reintento ("retry") = 7200 (2 horas)

Expiración ("expire") = 2592000 (30 días)

TTL mínimo ("minimum ttl") = 172800 (2 días)

En circunstancias especiales, por ejemplo, en caso de que se prevean cambios importantes o numerosos en la zona, los valores anteriores de "Refresco" y "TTL mínimo" pueden ser temporalmente modificados a la baja con la suficiente antelación al momento previsto para los cambios (48 horas antes), para que los cambios se propaguen en Internet con mayor celeridad (los secundarios se actualicen antes y los datos "cacheados" en otros servidores de DNS sean desechados antes). En estos casos, los valores mínimos recomendados son los siguientes:

Refresco ("refresh") = 14400 (4 horas)
Reintento ("retry") = 7200 (2 horas)
Expiracion ("expire") = 2592000 (30 dias)
TTL minimo ("minimum ttl") = 28800 (8 horas)

2.14.- En el caso de que un servidor del ES-NIC este haciendo de servidor secundario opcional para la zona de segundo nivel bajo "es" en cuestión, los valores normales y mínimos de los parámetros temporales del SOA mencionados en el apartado anterior no son "recomendados" sino obligatorios. En este caso, el poner valores inferiores a los mínimos obligatorios puede dar lugar a que el dominio sea dado de baja.

2.15.- El campo "mail address" del SOA de cualquier zona de DNS de segundo nivel bajo "es" debe contener siempre una dirección de correo electrónico (típicamente conocida como dirección de "hostmaster") válida y regularmente atendida por las personas de contacto técnico para el dominio en cuestión, con la que se pueda contactar en caso necesario. A la hora de especificar dicha dirección en el SOA de la zona hay que sustituir el carácter "@" por el carácter "." (por ejemplo, "hostmaster.dominio.es" para representar la dirección de e-mail "hostmaster@dominio.es").

2.16.- Puesto que es una practica frecuente en Internet intentar comunicar con la dirección de e-mail "postmaster@dominio" en caso de problemas de mensajería electrónica con dicho "dominio", se recomienda que para todo dominio de segundo nivel bajo "es" la dirección de e-mail "postmaster@dominio.es" exista y sea regularmente atendida por el responsable de mensajería electrónica de la organización titular del dominio.

3.- LECTURAS RECOMENDADAS:

- [1] RFC 1032, M. Stahl, "Domain Administrators Guide", Noviembre 1987
<ftp://ftp.rediris.es/docs/rfc/10xx/1032/>
 - [2] RFC 1033, M. Lottor, "Domain Administrators Operations Guide", Noviembre 1987
<ftp://ftp.rediris.es/docs/rfc/10xx/1033/>
 - [3] RFC 1034, P. Mockapetris, "Domain Names - Concepts and Facilities", Noviembre 1987
<ftp://ftp.rediris.es/docs/rfc/10xx/1034/>
 - [4] RFC 1035, P. Mockapetris, "Domain Names - Implementation and Specification", Noviembre 1987
<ftp://ftp.rediris.es/docs/rfc/10xx/1035/>
 - [5] RFC 1123, R. Braden, "Requirements for Internet Hosts - Application and Support", Octubre 1989.
<ftp://ftp.rediris.es/docs/rfc/11xx/1123/>
 - [6] RFC 1536, A. Kumar, J. Postel, C. Neuman, P. Danzig, y S. Miller, "Common DNS Implementation Errors and Suggested Fixes", Octubre 1993.
<ftp://ftp.rediris.es/docs/rfc/15xx/1536/>
 - [7] RFC 1537, P. Beertema, "Common DNS Data File Configuration Errors", Octubre 1993.
<ftp://ftp.rediris.es/docs/rfc/15xx/1537/>
 - [8] RFC 1713, A. Romao, "Tools for DNS debugging", Noviembre 1994.
<ftp://ftp.rediris.es/docs/rfc/17xx/1713/>
 - [9] RFC 1912, D. Barr, "Common DNS Operational and Configuration Errors", Febrero 1996.
<ftp://ftp.rediris.es/docs/rfc/19xx/1912/>
 - [10] P. Vixie, "Name Server Operations Guide for BIND", URL:
<http://www.dns.net/dnsrd/docs/bog/bog.html>
 - [11] P. Albitz y C. Liu, "DNS and BIND", Ed. O'Reilly & Associates, ISBN 1-56592-010-4, Julio 1994.
 - [12] ES-NIC, "Ejemplo de configuración del programa BIND", URL:
<ftp://ftp.nic.es/dns/bind.txt>
-

SERVICIO DNS

<http://www.ual.es/ServInf/Comunicaciones/dns.html>

El servicio DNS (Domain Name Server) traduce las direcciones IP a nombres de dominio internet. Este servicio a veces resulta necesario para poder acceder a ciertos recursos de la red (bases de datos, ftp anonymous, aplicaciones, ...).

Para darse de alta en este servicio es imprescindible obtener la dirección IP del equipo. Que según el sistema operativo se hace de la siguiente forma:

- Cualquier ordenador con una pegatina. En ella tiene apuntado la dirección IP.
- Windows 95/98: Menú Inicio / Ejecutar / winipcfg
- Windows NT 4.0: Menú Inicio / Configuración / Panel de control / Red / Protocolos

seleccionar el Protocolo TCP/IP y pulsar Propiedades.

- Windows 3.1: Abrir el trumpet y mirar en las frases escritas al principio la dirección IP.
- MS-DOS: Ejecutamos lo siguiente:
 - cd \nrsa
 - edit config.tel
 - mirar la dirección IP que viene al principio del fichero.

Una vez obtenida la dirección IP, llamamos por teléfono al 5544, 5743 o enviamos un correo electrónico a jose@ual.es, acasado@ual.es, dperez@ual.es, raul@ual.es, con la dirección IP y el nombre que le queramos dar al computador.

Servidor DNS

<http://www.redes.ccm.itesm.mx/internet/sdns.html>

Un servidor DNS (Domain Name System) se utiliza para proveer a las computadoras de los usuarios (clientes) un nombre equivalente a las direcciones IP. El uso de este servidor es transparente para los usuarios cuando este está bien configurado.

Cada LAN (Red de área local) debería contar con un servidor DNS. Estos servidores trabajan de forma jerárquica para intercambiar información y obtener las direcciones IP de otras LANs.

NIC (Network Information Center) es el organismo encargado de administrar el DNS a nivel mundial. NIC México se encarga de administrar todos los nombres (dominios) que terminen con la extensión mx. En la rectoría del sistema se encargan de administrar los nombres que terminen con itesm.mx. El Campus Ciudad de México cuenta con un Servidor DNS primario y un Servidor DNS secundario, los cuales mantienen las tablas de los nombres que terminen con ccm.itesm.mx.

Nuestro servidor DNS primario tiene la dirección 148.241.155.10 y el secundario 148.241.129.10. Para que la computadora funcione adecuadamente debes contar con estos valores. Si configuras adecuadamente el servicio de DHCP del campus automáticamente obtienes estos valores. Si cuentas con una dirección IP fija debes agregar manualmente estos valores. En el Intratec de alumnos obtiene automáticamente estos valores, pero en el Intratec de profesores hay que agregarlo a la hora de hacer configuración. Si se siguen adecuadamente las configuraciones no deberás tener problema. Configuración para dirección IP fija.

Configuración de Intratec

Configuración para DHCP

Intratec

<http://www.redes.ccm.itesm.mx/intratec/index.html>

Con este servicio puedes tener acceso a muchos servicios que se ofrecen en el Campus desde la comodidad de tu casa, esto es, puedes tener acceso a Lotus Notes (clases rediseñadas), correo electrónico, consultas en línea, inscripciones remotas, catálogo de biblioteca, etc. Para ello debes contar con una computadora en tu casa, un modem y una línea telefónica.

El objetivo de Intratec es ofrecerle a los alumnos los servicios de la IntraNet que se esta formando en el campus.

Requerimientos de conexión:

- Computadora PC 486 o superior, con 8 Mb de memoria como mínimo.
- Windows 95.
- Línea telefónica y un módem.
- Cuenta de correo electrónico del Campus Ciudad de México.
- Configurar adecuadamente tu computadora (Consulta el [manual de instalación para profesores o para alumnos](#)).

Si deseas shareware adicional al que tiene de fábrica el Windows, puedes bajarlo de nuestro servidor ftp anónimo.

El uso de este servicio esta regulado por las normas y lineamientos que se han definido para su correcta utilización, para cumplir con la nueva Misión del Sistema hacia el 2005, como es el uso eficiente de la información y las telecomunicaciones y la capacidad de aprender por cuenta propia.

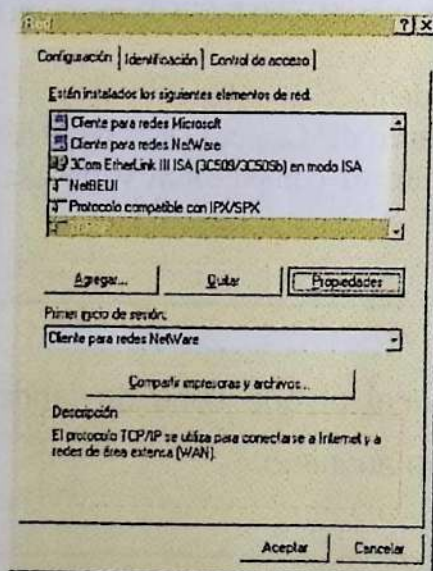
Configuración para DHCP

<http://www.redes.ccm.itesm.mx/dhcp/index.html>

Configuración TCP/IP dinámico para acceso a la red del campus.

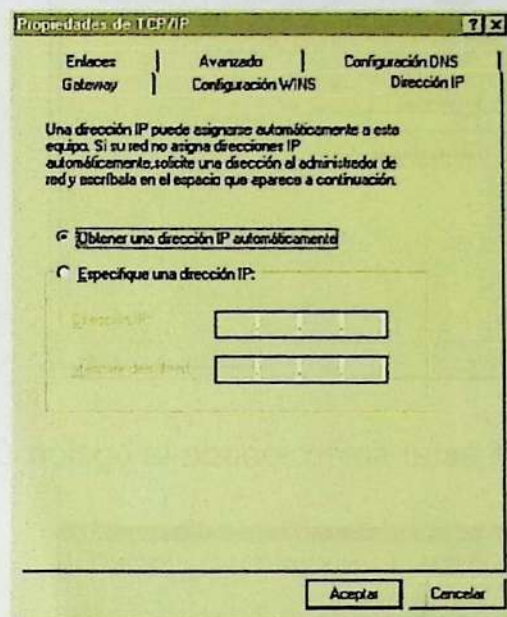
Para que puedas obtener una dirección IP en una computadora con Windows 95 o Windows 98 deberás tener la siguiente configuración:

Inicio → *Configuración* → *Panel de Control* → *Red*. Aparecerá una ventana con los protocolos y clientes instalados en tu computadora.

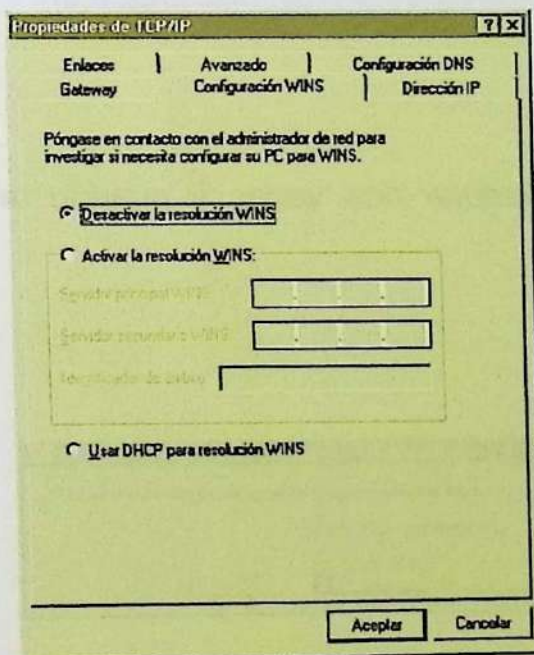


Debes hacer doble click en el protocolo TCP/IP. Si tienes un modem instalado debe ser el protocolo TCP/IP de la tarjeta de red de la computadora (No debe ser el que dice *Acceso telefónico a redes*).

En la pestaña de *Dirección IP* debe estar seleccionada la opción de *Obtener una dirección IP automáticamente*.



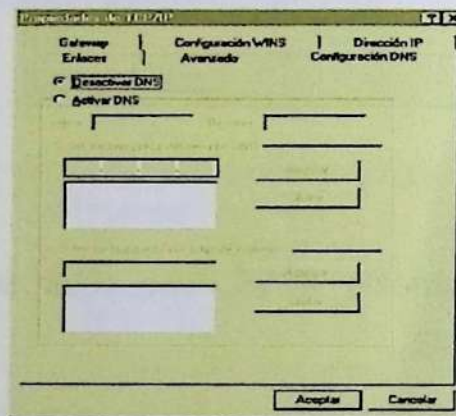
En la pestaña de *Configuración WINS* debe estar seleccionada la opción *Desactivar la resolución WINS*.



En la pestaña de *Puerta de Enlace* no debe existir ninguna dirección.



En la pestaña de *DNS* debe estar seleccionada la opción *Desactivar DNS*.



Para finalizar presiona *Aceptar* dos veces y cuando te pida que si deseas reiniciar, presiona *Sí*.

