



**UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES  
CUAUTITLAN**

**REDES DE COMPUTADORAS. INSTALACION Y  
CONFIGURACION DE UN SERVIDOR DE  
NOMBRES EN LINUX**

**TRABAJO DE SEMINARIO  
QUE PARA OBTENER EL TITULO DE:  
LICENCIADA EN INFORMATICA  
P R E S E N T A :  
ADRIANA ALVARADO HERNANDEZ**

**ASESOR: ING. JESUS MOISES HERNANDEZ DUARTE**

**CUAUTITLAN IZCALLI, EDO DE MEX.**

**1997.**

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

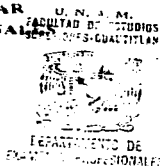
Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL  
AVENIDA LL  
MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN  
UNIDAD DE LA ADMINISTRACION ESCOLAR  
DEPARTAMENTO DE EXAMENES PROFESIONALES



DR. JAIME KELLER TORRES  
DIRECTOR DE LA FES-CUAUTITLAN  
PRESENTE.

ATN: ING. RAFAEL RODRIGUEZ CEBALLOS  
Jefe del Departamento de Exámenes  
Profesionales de la FES-C.

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlan, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario:

REPO. DE COMPUTACIONAL, INSTALACION Y CONFIGURACION DE UN SERVIDOR  
DE NOMBRES EN LINEA.

que presenta la presente ARAIANA ALVARADO HERNANDEZ  
con número de cuenta: 2000001200 para obtener el Título de:  
LICENCIADA EN INFORMATICA

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO.

ATENTAMENTE.

"POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlan Izcalli, Edo. de México, a 25 de AGOSTO de 19 87

MODULO:	PROFESOR:	FIRMA:
<u>III</u>	<u>ING. JOSÉ M. HERNÁNDEZ</u>	<u>[Firma]</u>
<u>III</u>	<u>M. I. ROSA PÉREZ VARGAS</u>	<u>[Firma]</u>
<u>II</u>	<u>ING. JESÚS ALVAREZ PARRA</u>	<u>[Firma]</u>

DEP/VOBOSER

*Agradecimientos*

*A mis padres*

*Por creer en mí, por su amor, confianza y apoyo en todo momento*

*A mis hermanos*

*Por darme su ayuda y cariño:*

*A Gonzalo*

*Por siempre brindarme su apoyo y ayuda.*

# INDICE

<b>INTRODUCCION.....</b>	<b>3</b>
<b>OBJETIVOS.....</b>	<b>5</b>
<b>1. INTRODUCCIÓN A LINUX.....</b>	<b>7</b>
1.1 DESCRIPCIÓN DE LINUX.....	7
1.2 HISTORIA DE LINUX.....	8
1.3 VERSIONES DE UNIX.....	8
1.4. VERSIONES DE LINUX.....	9
1.5 CARACTERÍSTICAS DE UNIX/LINUX.....	12
<b>2. INTRODUCCIÓN AL DNS.....</b>	<b>16</b>
2.1 EL TCP/IP DE INTERNET.....	16
2.2. CRECIMIENTO DE INTERNET.....	16
2.3 NOMBRES EN INTERNET.....	17
2.3.1 Dominios.....	18
2.3.2 Subdominios.....	18
2.3.3 Direcciones IP.....	18
2.4 NOMBRES DE DOMINIO EN INTERNET.....	20
2.5 NOMBRES PARA LAS MÁQUINAS.....	22
2.6 ¿PARA QUE SIRVE UN SERVIDOR DNS?.....	23
2.7 FUNCIONES DE UN DNS.....	23
2.8 RESOLUCIÓN DE NOMBRES DE DOMINIO.....	26
2.9 FORMATO DE LOS MENSAJES DEL SERVIDOR DE DOMINIOS.....	27
2.10 REQUERIMIENTOS DE HARDWARE Y SOFTWARE.....	31
<b>3 CONFIGURACIÓN DE UN DNS.....</b>	<b>34</b>
3.1. SERVICIO DE NOMBRE DE DOMINIO.....	34
3.2. CONFIGURACIÓN DEL RESOLVEDOR.....	35
3.2.1 El archivo <i>host.conf</i> .....	35
3.2.2 El archivo <i>resolv.conf</i> .....	37
3.3 CONFIGURACIÓN DE NAMED.....	38
3.4 DESEMPEÑO DEL CACHE.....	39
3.5 EL ARCHIVO NAMED.HOSTS.....	40
3.6 ARCHIVOS DE BASES DE DATOS Y REGISTRO DE RECURSO.....	42
3.7 EL ARCHIVO NAMED.LOCAL.....	44
3.8 EL ARCHIVO NAMED.HOSTS.....	45
3.9 EL ARCHIVO NAMED.REV.....	47
3.10 EL ARCHIVO NAMED.CA.....	48

<b>4 CASO PRÁCTICO. CONFIGURACIÓN DE UN SERVIDOR DE NOMBRES (DNS) DENTRO DE LA FACULTAD DE ESTUDIOS SUPERIORES CUAUTTLÁN.</b> .....	<b>51</b>
4.1 JUSTIFICACIÓN DE LA INSTALACIÓN.....	51
4.2 CONSIDERACIONES ADMINISTRATIVAS.....	51
4.3 INSTALACIÓN DEL SERVIDOR DE NOMBRES (DNS).....	55
4.4 CONFIGURACIÓN DE ARCHIVOS.....	56
4.5 SOLUCIÓN DE PROBLEMAS.....	60
4.6 ADMINISTRACIÓN DEL SERVIDOR DE NOMBRES.....	61
<b>5 CONCLUSIONES</b> .....	<b>62</b>
<b>BIBLIOGRAFIA</b> .....	<b>63</b>

## **INTRODUCCION.**

Para poder acceder a alguna computadora dentro de la red Internet, se debe conocer el nombre o su dirección numérica (IP) con la que se identifica cada una de las computadoras que se encuentran dentro de dicha red, sin embargo, con el crecimiento de esta red, se tuvo la necesidad de crear nuevos sistemas para la asignación de nombres de dominio. Estos nombres se especifican por medio de un sistema llamado servicio de nombres de dominio (DNS), el cual transforma nombres de máquinas en direcciones IP que puedan leer los usuarios de una red, esto con el fin de que el usuario le sea más fácil recordar el nombre de una máquina a la que desee acceder y para agilizar la comunicación a los diferentes dominios.

En el presente trabajo se describen las características más importantes de un Servidor de Nombres (DNS), así como la instalación y configuración utilizando el sistema operativo Linux con el principal objetivo de que la Facultad de Estudios Superiores Cuautitlán cuente con este servicio

Este trabajo se divide en cuatro capítulos:

En el capítulo 1 se ofrece un panorama general acerca de lo que es el sistema operativo Linux, como fue evolucionando y algunas de sus características generales como son los procesos de multitarea, multiusuario, programación de shells, etc.

El capítulo 2 se enfoca a algunos puntos básicos acerca de Internet, que es una red de comunicación mundial en la que se desempeña un Servidor de Nombres, además de los conceptos básicos de un DNS con el propósito de que el lector identifique la relación existente entre estos temas.

En el capítulo 3 se describen los procesos de configuración de los archivos de un Servidor de Nombres entre los que se encuentran los siguientes: *host.conf*, *resolv.conf*, *name.boot*, etc.

Por último en el capítulo 4 se tratará el caso práctico, en el que se instala y configura un Servidor de Nombres dentro de la Facultad de Estudios Superiores Cuautitlán, debido a que no cuenta con este servicio. Se tratan aspectos administrativos, de justificación de la implantación del Servidor de Nombres, así como la administración del mismo.



## **OBJETIVOS.**

### *General*

- Identificar las ventajas de un Servidor de Nombres (DNS) dentro de una red.

### *Específico*

- Instalar un Servidor de Nombres, con el fin de proporcionar este servicio dentro de la Facultad de Estudios Superiores Cuautitlán.
- Describir las características del hardware y software utilizado en la implementación de un Servidor de Nombres.

# **CAPITULO 1**

## **INTRODUCCION A LINUX**

## **1. INTRODUCCIÓN A LINUX.**

### ***1.1 Descripción de Linux.***

Linux es un sistema operativo gratuito para PC's con procesadores 386, 486 y pentium, este sistema ha sido desarrollado por una gran variedad de programadores en todo el mundo, con el fin de crear un sistema muy parecido a lo que es Unix; además ofrece algunas de las capacidades de Unix, como son la multitarea y el multiprocesamiento a un gran número de usuarios en plataformas compatibles con PCs IBM.

Obtenerlo gratuitamente en Internet es una buena ventaja ya que no se tiene que estar pagando cada determinado tiempo para su actualización, lo único que se debe hacer es acceder a la red y bajar el software actualizado, además se pueden obtener numerosas aplicaciones como hojas de cálculo, procesadores de textos, bases de datos y hasta juegos. En Linux puede tener acceso a su código fuente lo cual permite personalizar el sistema operativo de acuerdo a sus necesidades. Esto no es posible con otros sistemas operativos comerciales como Windows 95, OS/2, Novell, etc.

El hecho de que Linux sea gratuito y que ninguna firma comercial lo respalde puede representar una desventaja, ya que en el momento que al usuario de Linux le surga un problema, no existe alguna compañía especializada que de soporte a este sistema, sin embargo, si se desea aprender a utilizar Unix un buen comienzo es el sistema operativo Linux que proporciona al usuario el manejo a nivel de administrador de un sistema de red.

## **1.2 Historia de Linux.**

Linux fue creado en 1991 en Finlandia por un estudiante de informática llamado Linus Torvalds que esperaba crear una versión más sólida de Minix, este era un sistema operativo tutorial escrito por el profesor de informática Dr. Andrew Tannebaum.

Linus Torvalds proporcionó una plataforma mejor para los usuarios de Minix que pudiera ejecutarse en cualquier IBM PC, con procesadores 386.

Torvalds desarrolló la primera versión de Linux, sin embargo, actualmente un gran número de programadores en todo el mundo participan en el crecimiento y mantenimiento del sistema.

## **1.3 Versiones de Unix**

Son muchos y distintos los distribuidores que ofrecen las versiones de Unix, pero pueden dividirse en dos grandes categorías: el System V y la BSD (Berkeley Software Distribution) de Unix. El sistema V es la versión desarrollada por los creadores de Unix AT&T Bell Laboratories. La BSD de Unix estuvo relacionada con la versión de Unix que desarrolló y proliferó en la Universidad de California en Berkeley. La versión más popular la vendió Sun Microsystems para sus estaciones de trabajo.

Para la versión 4 del System V de Unix, AT&T y Sun Microsystems fusionaron sus dos versiones en el System V. Debido a esta fusión se le considera a este último el sistema estándar de Unix.

Unix System Laboratories creó la versión 4.2 del System V (SVR4.2, ahora pertenece a Novell inc.). Esta versión combina la potencia y las capacidades de Unix con una interfaz gráfica para usuarios. En este sistema operativo puede navegar

con facilidad un usuario inexperto en computación o lo pueden personalizar los administradores de sistemas u otros expertos. Novell vende el SVR4.2 como Unix Ware.

Antes de la introducción al mercado del SVR4.2, los usuarios de Unix sólo veían un indicador de líneas de comandos (un signo de pesos) luego de conectarse al sistema. Sin ninguna guía en la pantalla, los usuarios inexpertos a menudo se desalentaban tanto que preferían abandonar Unix y utilizar otro sistema. Esto contribuyó a que Unix se ganara durante años la reputación de ser un sistema operativo poco amistoso. Sin embargo, la experiencia y los comentarios favorables hicieron que los usuarios que lo consideraban poco amistoso disiparan sus opiniones y con esto la reputación de Unix cambió lentamente.

#### **1.4. Versiones de Linux.**

A partir de la versión 1.0 que data de marzo de 1994, Linux ya no es una versión de prueba (versión beta). Aún hay errores en el sistema y nuevos errores irán apareciendo y serán solucionados con el tiempo. Como Linux es un proyecto de desarrollo abierto, todas las nuevas versiones que vayan apareciendo estarán disponibles para el público, sean o no estables. Sin embargo, para ayudar a diferenciar al usuario si una versión es o no estable se ha establecido lo siguiente: las versiones 1.x.y en donde x sea par son versiones estables y el incremento de y implica la corrección de algún error. Por lo tanto de la versión 1.2.2 a la 1.2.3 sólo hay corrección de errores, pero ninguna característica nueva. Las versiones 1.x.y con x impar son betas para los desarrolladores, pudiendo ser inestables y están siendo ampliadas continuamente con nuevas posibilidades. Las actuales versiones de desarrollo son las numeradas 1.3.x, y la más reciente versión estable es la 1.2.13.

Algo a tener en cuenta es que Linux está desarrollado siguiendo un modelo abierto y distribuido, en lugar de uno cerrado y centralizado como la mayor parte del software. Esto significa que la versión actualmente en desarrollo es siempre pública (con un retraso de una o dos semanas) para que cualquiera pueda usarla. El resultado es que en cualquier momento que se añada una nueva funcionalidad y salga a la luz la nueva versión, ésta casi siempre tendrá errores, pero serán detectados y corregidos rápidamente, a menudo en cuestión de horas, ya que mucha gente trabaja en ello.

En contraste, el modelo centralizado y cerrado significa que hay sólo una persona o un equipo trabajando en el proyecto, y sólo publican software que ellos piensan que esté trabajando bien. A menudo esto conlleva largos periodos de tiempo entre versiones, largas esperas para la corrección de errores y un desarrollo más lento. Por supuesto que la última versión de este tipo de software es a menudo de mejor calidad para el público, pero la velocidad de desarrollo es normalmente mucho más lenta.

Al 8 de febrero de 1996, la actual versión estable de Linux es 1.2.13, y el nuevo árbol de desarrollo, 1.3.x, está actualmente por la versión 1.3.60.

### ***Distribuciones***

El único elemento común a todas las versiones Linux es su *kernel* (núcleo del sistema operativo), que se desarrolla de forma coordinada y con actualizaciones sistemáticas. Sin embargo todo sistema operativo necesita, junto al núcleo del sistema, todo un conjunto de utilidades y herramientas de instalación, configuración y uso. Ahí juegan su papel las diferentes distribuciones: algunos particulares, entidades y empresas se dedican a hacer determinadas recopilaciones de software que ellos mismos preparan para que sean fácilmente instalables y configurables. Todas ellas facilitan el software junto a su código fuente, pero la flexible licencia

GNU (Licencia Pública General) a la que se apegan permite tanto ofrecerlas gratuitamente como distribuir las por canales comerciales (lo que se paga es el trabajo de recopilación, el software de cosecha propia que pueda aportar, una presentación más elaborada, gastos de distribución y soporte técnico al usuario).

Algunas de las distribuciones más conocidas son:

#### *Slackware*

Esta distribución es quizás de las más extendidas en todo el mundo y la más conocida. Incluye todo el software que cualquier usuario medio o avanzado pueda necesitar, y su método de instalación es accesible aunque lejos de las típicas y vistosas instalaciones a las que el usuario está acostumbrado en los sistemas comerciales. Tiene diversos derivados comercializados bajo diferentes nombres (por ejemplo "Slackware Pro") que mejoran la instalación e incluyen manuales y bibliografía en un único lote. Está en todos los FTPs principales, siendo <http://www.cdrom.com/> su distribuidor de origen.

#### *Red Hat*

Creada por Red Hat Software, en Connecticut, EE.UU. Una de sus ventajas es el atractivo sistema de instalación (en modo gráfico) y el cómodo mantenimiento de componentes de software, lo que facilita enormemente las tan frecuentes actualizaciones. Se puede obtener tanto gratuitamente en la red como adquiriendo el CDROM correspondiente. Otras empresas comercializan también sistemas basados en Red Hat, como Caldera Inc. y Pacific Hi-Tech. Sus creadores están en <http://www.redhat.com>. El mencionado sistema de gestión de componentes de software es obra suya, pero lo han ofrecido con carácter abierto y gratuito a los demás desarrolladores bajo la licencia de GNU, por lo que es previsible que en el

futuro otros muchos asuman este sistema en sus propias distribuciones, lo que facilitará enormemente las actualizaciones.

### *Debian*

La Free Software Foundation (FSF) es bien conocida entre los usuarios de software gratuito para Unix. Son los creadores del sistema GNU, su futuro Unix gratuito. Ya hay mucho material pero no un sistema operativo completo, así que mientras tanto ofrecen un Unix integrado por el núcleo de Linux y el software de GNU. <http://www.debian.org>.

### **1.5 Características de Unix/Linux.**

El sistema operativo Linux es muy parecido a Unix, como ya se mencionó, por lo que ambos cuentan con las siguientes características :

#### ***Multitarea.***

La multitarea es la ejecución de varios procedimientos al mismo tiempo.

Linux ofrece la realización de multitareas preferentes, a diferencia de otros sistemas como MS-DOS, Windows 95 que realizan multitareas cooperativas.

En las multitareas preferentes cada programa no se ejecuta sino hasta que el sistema operativo lo aparta para permitir que otros programas corran y en las multitareas cooperativas los programas corren hasta que ellos mismos permiten la ejecución de otro programa o no tienen otra cosa que hacer durante ese periodo.



### *Multiusuario.*

El término multiusuario se refiere a una computadora compartida por varios usuarios. Linux permite que más de un usuario pueda trabajar simultáneamente en una misma aplicación, desde las mismas terminales o desde terminales separadas.

### *Shells*

Los shells son programas diseñados para aceptar comandos y ejecutarlos, en casi todas las versiones de Linux existen diversos shells estándares disponibles. El shell de Linux funciona como interprete entre el usuario y el kernel (Linux). Esta característica se emplea con el fin de personalizar el sistema y hacerlo más amigable para los usuarios.

### *Independencia de dispositivos bajo Unix/Linux.*

Unix toma a los periféricos como archivos, si se necesita agregar nuevos dispositivos el administrador del sistema añade al kernel el enlace apropiado, es decir que cada vez que se solicite el servicio de algún dispositivo este se fusionará con el kernel garantizando así el buen funcionamiento de los dispositivos.

Debido a que Linux es un clon de Unix, comparte algunas de las ventajas en la independencia de dispositivos, por lo tanto Linux también cuenta con un kernel, sin embargo debido a que Linux no es respaldado por un marca comercial y puesto que fue desarrollado por diferentes programadores que no tenían acceso a todos los equipos creados para IBM PC, Linux no es compatible con todo el hardware para PC.

### *Comunicaciones y capacidades en red.*

La capacidad y flexibilidad de Unix y Linux en las comunicaciones es superior a otros sistemas operativos como Windows, Novell, etc., ya que cuenta con una amplia gama de aplicaciones y programas para establecer comunicaciones entre redes.

Unix y Linux cuentan con el software necesario para transferir archivos extensos que se encuentran en otro sistema en el otro extremo del país, además proporciona utilerías para la transferencia de mensajes, entre otras cosas.

### *Portabilidad de sistemas abiertos.*

Es la capacidad de adaptabilidad de un sistema operativo al ser transportado a una plataforma diferente y que su funcionamiento sea el mismo. Unix es un sistema operativo portátil, por lo que proporciona el medio para que varias plataformas que ejecutan Unix establezcan una comunicación en forma precisa y eficaz con cualquiera de las demás sin agregar ninguna interfaz de comunicaciones especial.

# **CAPITULO 2**

## **INTRODUCCION AL DNS**

## **2. INTRODUCCIÓN AL DNS.**

### ***2.1 El TCP/IP de Internet.***

Debido a la importancia y potencial de la tecnología de red de redes desde hace varios años las agencias gubernamentales de los Estados Unidos han proporcionado fondos para la investigación realizada por la Agencia de Proyectos de Investigación Avanzada (ARPA, por sus siglas en inglés). La tecnología ARPA incluye un grupo de estándares de red que especifican los detalles de como se comunican las computadoras, así como un grupo de reglas para interconectar redes y para rutear el tráfico. Conocido de manera oficial como el grupo de protocolos de Internet TCP/IP (Protocolo de Control de Transmisiones/Protocolo Internet), éste puede utilizarse para comunicarse a través de cualquier grupo de redes interconectadas. Por ejemplo, algunas empresas utilizan el TCP/IP para interconectar todas las redes dentro de la corporación, aún cuando las empresas no tengan conexión hacia redes externas.

La tecnología TCP/IP es especialmente interesante debido a que su viabilidad ha sido demostrada a gran escala. Esta tecnología forma la base para una red de redes globales que conecta hogares, campus universitarios y otras escuelas, corporaciones y laboratorios en muchos países.

### ***2.2. Crecimiento de Internet.***

Internet es la red de redes de computadoras que se comunican entre sí para intercambiar información. Debido a la popularidad de esta red el número de computadoras a aumentado de manera constante.

El crecimiento en la red Internet ha sido muy significativo en los últimos diez años. En 1984 había alrededor de 1000 computadoras conectadas a Internet. Cinco años después, en 1989 aumentó a 100 000. En 1992 el número de computadoras era de 1000 000 . Para 1994, existían más de 3.2 millones de computadoras en la red con un número aproximado de 20 millones de usuarios.

Con las nuevas herramientas que proporciona Internet como Word Wide Web (WWW) y mosaic, el acceso y la búsqueda de información es más sencilla por lo que muchos usuarios no técnicos están comenzando a hacer uso de esta red.

Actualmente la mayoría de las naciones cuenta de alguna forma con una conexión a Internet y se siguen integrando nuevas computadoras y redes.

### **2.3 Nombres en Internet.**

Para poder interactuar con una computadora dentro de una red es necesario saber su nombre. En Internet existen millones de computadoras y cada una de ellas debe tener un nombre único asignado, estos nombres se especifican por medio de una convención llamada Servicio de Nombres (DNS), que se determinan con los números 1032,1033,1034 y 1035 en las peticiones para comentarios de Internet (RFC).

El formato utilizado en el sistema de nombres de dominio es el siguiente:

*subdominio . subdominio . dominio*

En donde los subdominios y dominios se listan, separados por un punto, de acuerdo a diferentes niveles de especificación.

### 2.3.1 Dominios.

En el sistema DNS el registro de direcciones se lleva a cabo tomando en cuenta las jerarquías de las redes o computadoras a registrar. La parte del nombre al extremo derecho, el campo de dominio, proporciona la categoría más general.

Los últimos campos de una dirección DNS, son los de más alto nivel e indican el género del usuario y su ubicación en un país. A excepción de Estados Unidos, cada nación cuenta con un dominio formado por dos letras especificadas en base en su código de país, esto se encuentra en el documento 3166 de la Organización de Estándares Oficiales (ISO).

### 2.3.2 Subdominios.

Dentro de un nombre de DNS los campos de subdominio identifican una computadora o una dirección específica dentro de un dominio. En un nombre de dominio los subdominios son más específicos de derecha a izquierda. El campo de subdominio que se encuentra en el lado derecho, junto al campo de dominio indica por lo general una organización particular dentro de un dominio determinado. Los campos de subdominios secundarios indican un departamento, grupo o computadora en una organización. Este agrupamiento de subdominios produce una estructura lógica de árbol dentro del dominio.

Se usa el símbolo @ para especificar a un usuario, alias o buzón en el nombre DNS del sitio.

### 2.3.3. Direcciones IP.

Para la conexión a los sitios IP en Internet se usa un conjunto de protocolos TCP/IP para facilitar la comunicación.

Cada anfitrión es identificado por TCP/IP con un número único, conocido como su dirección IP. La dirección IP tiene un valor de 32 Bits y se divide en cuatro octetos<sup>1</sup>, separados por puntos.

Cada computadora tiene una sola dirección IP única para cada interfaz física que tiene conectada a la red. Una dirección IP se divide en dos partes: la primera especifica una red particular y la segunda es la sección de anfitrión que identifica alguna máquina en la red. Existen cinco categorías de direcciones IP de acuerdo al tipo de dirección de red y se conocen como Clase A hacia Clase E.

Las direcciones de la Clase A han sido utilizadas por las grandes corporaciones y proveedores de red. En la Clase A, el primer octeto tiene un valor de 1 a 126 y la sección de la red consta del primer octeto. La cantidad de redes asignadas a las Clase A es de 126, y cada red puede tener más de 16 millones de computadoras.

Las direcciones de Clase B se asignan a grandes compañías y organizaciones, como las universidades. En la Clase B se utilizan los primeros dos octetos para identificar la sección de la red y tienen el primer octeto en el rango de 128 a 191. El total de redes en esta clase es de 16 382 cada una con un número de 65 534 computadoras.

Por lo general las redes de la Clase C son asignadas a negocios u organizaciones pequeños. En esta clase se usan tres octetos para especificar la sección de red y el primer octeto está en el rango de 192 a 233. La Clase C proporciona mas de dos millones de redes, pero con solo 254 anfitriones por red.

La clase D se utiliza para transmisiones de emisión múltiple.

---

<sup>1</sup> En terminología en redes, un campo de 8 bits se conoce como un octeto en lugar de un byte. Con esto se evitan confusiones ya que, algunas computadoras utilizan tamaños de byte diferentes de 8 bits.

Las direcciones de Clase E no se utilizan, están reservadas para un uso futuro.

#### **2.4 Nombres de dominio en Internet.**

El estándar de nombres de dominio especifica un espacio de nombre jerárquico abstracto con valores arbitrarios para los nombres. Como el sistema de dominio establece sólo la forma de los nombres y no sus valores actuales, es posible, para cualquier grupo que constituya una instancia de sistema de dominio seleccionar nombres para todas las partes de su jerarquía. Sin embargo, la mayoría de los usuarios de tecnología de dominio sigue la jerarquía de nombres utilizada por el sistema de dominio oficial de Internet, ya que el esquema de Internet es completo y flexible. Se puede adaptar a una amplia variedad de organizaciones y permite a cada grupo seleccionar entre una jerarquía de nombres asignada geográficamente o una función de la estructura organizativa. Además la mayor parte las localidades sigue el esquema de Internet porque de esta manera puede conectar sus instalaciones TCP/IP a la red Internet sin cambiar nombres.

En la tabla 2.1 se listan los dominios de Internet

<b>NOMBRE DE DOMINIO</b>	<b>SIGNIFICADO</b>
COM	Organizaciones comerciales
EDU	Instituciones educativas
GOV	Instituciones gubernamentales
MIL	Grupos militares
NET	Centros mayores de soporte de red
ORG	Organizaciones diferentes a las anteriores
ARPA	Dominio temporal de ARPANET (obsoleto)
INT	Organizaciones internacionales
Código de país	País en particular (según esquema geográfico)



El nombre de nivel superior permite dos jerarquías de nombres completamente diferentes: el esquema geográfico y el organizacional. El geográfico divide a las máquinas por país . El dominio de nivel superior también permite que las organizaciones se agrupen en función de su organización. Cuando una organización desea participar en el sistema de nombres de dominio, decide la forma en que desea que se registre y solicita su aprobación. La autoridad central revisa la solicitud y asigna un subdominio a la organización bajo uno de los dominios de nivel superior existentes. Por ejemplo, es posible que una universidad se registre con un dominio de segundo nivel EDU o que se registre según el estado o país en que se localiza.

Por ejemplo, una máquina llamada *peanut* en el departamento de ciencias computacionales de la Universidad de Purdue tiene el siguiente nombre de dominio oficial:

*peanut.cs.purdue.edu*

El nombre de la máquina fue aprobado y registrado por el administrador de red local en el Departamento de Ciencias Computacionales. El administrador del departamento había obtenido previa autorización para el subdominio *cs.purdue.edu* de una autoridad de la red universitaria, quien a su vez obtuvo permiso para administrar el subdominio *purdue.edu* de la autoridad de Internet. La autoridad de Internet conserva el control del dominio, por lo que nuevas universidades pueden añadirse sólo con su permiso. De igual forma, el administrador de red de la Universidad de Purdue conserva la autoridad para el subdominio *purdue.edu*, de manera que los nuevos dominios de tercer nivel sólo pueden ser añadidos con la autorización del administrador.

## **2.5 Nombres para las máquinas.**

Los primeros sistemas de computadoras forzaban a los usuarios a entender direcciones numéricas para objetos como tablas de sistema y dispositivos periféricos. Los sistemas de tiempo compartido mejoraron el cómputo al permitir que los usuarios inventaran nombres simbólicos y significativos para objetos físicos (Por ejemplo dispositivos periféricos) y objetos abstractos (por ejemplo, archivos). Un modelo similar apareció en las redes de computadoras. Los primeros sistemas soportaban conexiones punto a punto entre computadoras y utilizaban direcciones de hardware de bajo nivel para especificar máquinas. El enlace de redes introduce el direccionamiento universal así como el software de protocolo para transformar direcciones universales en direcciones de hardware de bajo nivel. Como en la mayor parte de los ambientes de computación hay varias máquinas, los usuarios necesitan nombres simbólicos y significativos para nombrarlas. Los usuarios preferían nombres para las máquinas a las incómodas direcciones de hardware.

Un nombre es sólo un identificador que consiste en una secuencia de caracteres seleccionados de un alfabeto finito. Los nombres sólo son útiles si el sistema puede transformarlos de manera eficiente para referirse al objeto que denotan. Una dirección IP se conoce como un nombre de bajo nivel y los nombres como de alto nivel.

La forma de los nombres de alto nivel es importante pues determina cómo son traducidos los nombres a nombres de bajo nivel o como conducen objetos, también determinan la forma en que se autoriza la asignación de nombres. Cuando sólo se tiene algunas máquinas interconectadas, la selección de nombres es fácil. En Internet, donde existen millones de máquinas conectadas, la selección de nombres es difícil. A mediados de 1986, la lista oficial de anfitriones de Internet contenía 3100 nombres registrados y 6500 alias.

## **2.6 ¿Para que sirve un servidor DNS?**

El mecanismo que implanta una jerarquía de nombres de máquina para las redes TCP/IP se conoce como Sistema de Nombre de Dominio (DNS). El DNS convierte direcciones IP en nombres que representan computadoras, redes y alias de correo, para que sea más fácil para las personas identificar una computadora y de esta manera facilitar la comunicación en la red.

El DNS usa una arquitectura cliente/servidor para mantener y distribuir nombres de dominio y direcciones IP corriendo sobre redes de área local, así como en Internet.

## **2.7 Funciones de un DNS.**

Como ya se mencionó, cuando las computadoras se comunican al usar TCP/IP utilizan la dirección IP numérica. Los nombres DNS son un dispositivo para identificar cuales son los anfitriones y a que red se está conectado. En los inicios de Internet la cantidad de anfitriones era pequeña, por lo que era sencillo tener una lista de los nombres y direcciones de cada anfitrión en una lista de archivos locales, pero con el tiempo esto llegó a ser pesado y lento, ya que cuando se integraba un nuevo anfitrión se tenían que actualizar todos los archivos de anfitriones en todas las computadoras. A medida que fue creciendo Internet el tamaño de los archivos aumento y se tuvo que crear un nuevo sistema para hacer la transformación de nombres DNS en direcciones IP por medio de una base de datos distribuida y un software que realiza la búsqueda y esta constituido por: espacio de nombres de dominio, servidores de nombre, resolvers.

### ***Espacio de nombres de dominio.***

El espacio de nombres de dominio es un estructura de árbol en donde cada nodo del árbol contiene una base de datos de información acerca del anfitrión bajo su autoridad, es decir es una lista de todos los nombres, direcciones IP, alias de correo etc., que se encuentren disponibles para consultarse en el sistema DNS.

### ***Servidores de nombres.***

Son programas que mantienen y guardan los datos ubicados en el espacio de nombres de dominio. Además, un Servidor de Nombres tiene toda la información para su área de autoridad, que se clasifican en zonas y se dividen entre diferentes servidores de nombres con el fin de proporcionar servicio redundante para una zona.

Como cada Servidor de Nombres tiene contacto con los demás servidores de nombres cuando hay una petición de información para la zona de la que es responsable un Servidor de Nombres determinado, el Servidor de Nombres regresa la información pero, si la petición de información era en una zona diferente, el Servidor de Nombres contacta al servidor indicado y contesta la petición.

### ***Resolutores.***

Son programas que proporcionan información de los servidores de nombre cuando se realiza una consulta acerca de un anfitrión en el espacio de nombres de dominio.

### ***Localización de Información.***

Para llevar a cabo la localización de información se utilizan dos programas : el *whois* y *nslookup*.

### *Whois.*

Hace una consulta al sistema DNS y regresa información sobre números de dirección de red, datos para contactos administrativos y técnicos, números telefónicos y direcciones postales y de correo electrónico. La sintaxis es la siguiente:

*whois* [nombre]

En donde *nombre* es una red o nombre de dominio de un usuario registrado

### *Nslookup.*

Consulta al sistema DNS y regresa información acerca de redes y anfitriones específicas. *Nslookup* regresa varios tipos diferentes de información, pero su comportamiento por omisión es resolver un nombre DNS y regresar la dirección IP correspondiente.

Este comando tiene dos formas básicas: interactiva y no interactiva. La forma interactiva tiene la siguiente forma:

*nslookup* [nombredeanfitrión]

El modo interactivo se utiliza cuando no se le dan argumentos o cuando el primer argumento comienza con un guión y el segundo argumento es el nombre o dirección IP de un nombre de servidor. En el modo interactivo, se puede personalizar el comportamiento DNS *nslookup*, al especificar el nombre del servidor a consultar, cambiar el nombre del dominio por omisión, etc. Existen varias opciones en el modo interactivo, pero se usa sobre todo para buscar información distinta de la dirección IP.

## **2.8 Resolución de nombres de dominio.**

La resolución de nombres de dominio se realiza de arriba hacia abajo (dentro de la jerarquía) , comenzando con el servidor de nombre raíz y siguiendo luego hacia los servidores localizados hacia abajo. Hay dos formas de utilizar el sistema de nombres de dominio: contactar un Servidor de Nombres cada vez o solicitar al sistema de servidores de nombres que realice la traducción completa. En cada caso, el software cliente forma una solicitud de nombres de dominio que contiene el nombre a resolver, una declaración sobre la clase de nombre, el tipo de respuesta deseada y un código que especifica si el Servidor de Nombres debe traducir el nombre completamente y se envía la solicitud a un Servidor de Nombres para su resolución.

Cuando un Servidor de Nombres de dominio recibe una solicitud, verifica si el nombre señala un subdominio sobre el cual tenga autoridad. Si esto es así, traduce el nombre a una dirección de acuerdo a su base de datos y anexa una respuesta a la solicitud, antes de enviarla de regreso al cliente. Si el Servidor de Nombres no puede resolver el problema del nombre completamente, verifica qué tipo de interacción especificó el cliente. Si el cliente solicita una traducción completa<sup>2</sup>, el servidor se pone en contacto con un Servidor de Nombres de dominio que pueda resolver el problema del nombre y devuelve la respuesta al cliente. Si el cliente solicita una resolución no recursiva (resolución iterativa), el Servidor de Nombres no puede dar una respuesta. Se genera una réplica que especifica el nombre del servidor que el cliente deberá de contactar la próxima vez para resolver el problema del nombre.

Un cliente debe saber cómo contactar al último servidor de nombre. Para asegurarse de que el Servidor de Nombres de dominio puede alcanzar a otros, el

---

<sup>2</sup> Una resolución recursiva en la terminología de nombre de dominio

sistema de dominio requiere que cada servidor conozca la dirección de un servidor para el dominio de un nivel inmediatamente superior (llamado padre).

Los servidores de nombres de dominio utilizan un puerto de protocolo bien conocido para toda comunicación así, los clientes saben cómo comunicarse con un servidor una vez que conocen la dirección IP de la máquina que se conecta al servidor. No hay una forma estándar de que los anfitriones localicen una máquina en el entorno local, el cual corre un servidor de nombre; esto se encuentra abierto para quien diseña el software cliente.

### ***2.9 Formato de los mensajes del servidor de dominios.***

Cuando un usuario invoca un programa de aplicación y proporciona el nombre de la máquina con la que la aplicación debe comunicarse, antes de poder utilizar protocolos como el TCP o el UDP para comunicarse con la máquina especificada, el programa de aplicación debe encontrar la dirección IP de la máquina. Debe pasar el nombre de dominio a la máquina local capaz de resolver el nombre y solicitar una dirección IP. El solucionador local verifica su memoria inmediata y devuelve la respuesta si hay alguna presente. Si el solucionador local no tiene una respuesta, formatea un mensaje y lo envía al servidor (esto es, se convierte en un cliente). El formato de mensaje permite a un cliente hacer varias solicitudes en un solo mensaje. Cada uno consiste en un nombre de dominio para el que el cliente busca una dirección IP, una especificación de la plaza de solicitud (es decir, una red de redes) y el tipo de objeto deseado (esto es, una dirección). El servidor responde con la devolución de un mensaje similar que contiene respuestas a las solicitudes para las que el servidor tiene asignaciones. Si el servidor no puede responder a todas las preguntas, la respuesta contendrá información acerca de otro Servidor de Nombres que el cliente puede contactar para encontrar la respuesta.

Las respuestas también contienen información acerca de servidores que están autorizados para responder y las direcciones IP de tales servidores. A continuación se muestra un formato de un mensaje de servidor de nombre de dominio en la tabla 2.2.

**Tabla 2.2 Formato de un mensaje de servidor de nombre de dominio**

0	16	31
<b>IDENTIFICACIÓN</b>		<b>PARAMETRO</b>
<b>NUMERO DE SOLICITUDES</b>		<b>NUMERO DE RESPUESTAS</b>
<b>NUMERO DE AUTORIDAD</b>		<b>NUMERO ADICIONAL.</b>
<b>SECCIÓN DE SOLICITUDES</b>		
...		
<b>SECCIÓN DE RESPUESTAS</b>		
...		
<b>SECCION DE AUTORIDAD</b>		
...		
<b>SECCION DE INFORMACION ADICIONAL</b>		
...		

Cada mensaje comienza con un encabezado fijo, que contiene el campo único IDENTIFICATION (IDENTIFICACION) que el cliente utiliza para confrontar las respuestas solicitadas y el campo PARAMETER (PARAMETRO) que especifica la operación solicitada y el código de respuesta. La tabla 2.3 proporciona la interpretación de los bits en el campo PARAMETER:



<b>Tabla 2.3 Significado de los bits del campo PARAMETER</b>	
<b>Bit del campo PARAMETRO</b>	<b>Significado</b>
0	Operación: 0 solicitud. 1 respuesta.
1-4	Tipo de solicitud: 0 estándar. 1 inversa. 2 terminado 1 (obsoleto). 3 terminado 2 (obsoleto).
5	Activado si se tiene una respuesta autorizada.
6	Activado si el mensaje está truncado.
7	Activado si se desea recursión.
8	Activado si la recursión está disponible.
9-11	Reservado
12-15	Tipo de respuesta: 0 sin error. 1 error de formato en la solicitud. 2 falla en el servidor. 3 el nombre no existe

El campo NUMBER OFF (NUMERO DE) proporciona un conteo de las entradas en la sección correspondiente que se presentan en el último mensaje. El campo NUMBER OFF QUESTIONS (NUMERO DE SOLICITUDES) proporciona el conteo de entradas que aparecen en la QUESTION SECTION (SECCION DE SOLICITUDES) del mensaje.

**QUESTION SECTION** contiene las solicitudes para las que se desea una respuesta. El cliente llena sólo la sección de solicitud; el servidor devuelve la solicitud y la respuesta en su replica. Cada solicitud consiste en un **QUERY DOMAIN NAME** (SOLICITUD DE NOMBRE DE DOMINIO) seguido por los campos **QUERY TYPE** (TIPO DE SOLICITUD) y **QUERY CLASS**, (CLASE DE SOLICITUD). El siguiente es un formato de la entrada de información en la sección de solicitud como se muestra en la tabla 2.4.

**Tabla 2.4. Formato de entrada de información en QUESTION SECTION**

0	16	31
SOLICITUD DE NOMBRE DE DOMINIO		
...		
TIPO DE SOLICITUD	CLASE DE SOLICITUD	

El nombre de dominio tiene una longitud variable. El cliente llena la solicitud; el servidor la devuelve junto con la respuesta.

Aún cuando el campo **QUERY DOMAIN NAME** tiene una longitud variable, la representación interna de los nombres de dominio hace posible, para el receptor conocer la longitud exacta. El **QUERY TYPE** codifica el tipo de solicitud (por ejemplo, si la solicitud se refiere a un nombre de máquina o a una dirección de correo). El campo **QUERY CLASS** permite que los nombres de dominio se utilicen para objetos arbitrarios debido a que los nombres oficiales de Internet son sólo de una clase. Los mensajes hacia o desde un Servidor de Nombres pueden contener un número impar de octetos.

En un mensaje de Servidor de Nombres de dominio, cada uno de los campos **ANSWER SECTION**, (SECCION DE RESPUESTAS) **AUTHORITY SECTION** (SECCION DE AUTORIDAD) y de la **ADDITIONAL INFORMATION SECTION** (SECCION DE INFORMACION ADICIONAL) consiste en un conjunto de registros de

recursos que describen los nombres de dominio y las transformaciones. Cada registro de recurso describe un nombre. Se muestra un formato en la tabla 2.5.

**Tabla 2.5 Formato de un registro de recurso**

0	16	31
RECURSO DE NOMBRE DE DOMINIO		
...		
TIPO	CLASE	
TIEMPO LIMITE DE DURACION	LONGITUD DE DATOS DE RECURSO	
DATOS DE RECURSO		
...		

En el campo RESOURCE DOMAIN NAME (RECURSO DE NOMBRE DE DOMINIO) contiene el nombre de dominio al que este registro de recursos se refiere. El campo TYPE (TIPO) especifica el tipo de datos incluidos en el registro de recurso; el campo CLASS (CLASE) especifica la clase de datos; el campo TIME TO LIVE (TIEMPO LIMITE DE DURACION) contiene un entero que especifica en número de segundos que la información en este registro de recursos se mantendrá en memoria inmediata. Esta es utilizada por clientes que han solicitado la asignación de un nombre y desean capturar el resultado. Los dos últimos campos contienen el resultado de la asignación, con el campo RESOURCE DATA LENGTH (LONGITUD DE DATOS DE RECURSOS) especificando el conteo de octetos en el campo RESOURCE DATA (DATOS DE RECURSO).

### 2.10 Requerimientos de hardware y software.

Los requerimientos de software son los siguientes:

Sistema operativo Linux Red Hat 4.2

El programa demonio *named*.

Los requerimientos de hardware son los siguientes:

<b>CPU</b>	<b>386/486/Pentium</b>
<b>BUS</b>	<b>ISA, EISA, PCI o VLB. MCA (bus propietario de IBM)</b>
<b>RAM</b>	<b>16 MB de RAM.</b>
<b>Espacio en disco</b>	<b>1 GB en disco duro.</b>
<b>Monitores</b>	<b>CGA, EGA, VGA y SVGA.</b>
<b>Ratones</b>	<b>Cualquier ratón estándar; por ejemplo, logitech, Microsoft y Mouse Systems.</b>
<b>Unidad de CD-ROM</b>	<b>Es altamente recomendable. Casi mejor habría que decir que imprescindible, dado que las distribuciones vienen en ese formato. En la actualidad, la mayoría de ellas están soportadas por el Linux. Asimismo tiene soporte para módem y tarjeta de sonido.</b>

**CAPITULO 3**  
**CONFIGURACION DE UN**  
**DNS**

### 3 CONFIGURACIÓN DE UN DNS.

#### 3.1. Servicio de nombre de dominio.

El DNS es implementado por el software Berkeley Internet Name Domain (BIND), que es un sistema cliente/servidor. Por el lado del cliente el BIND es usado por una biblioteca llamada el resolvidor, este genera las preguntas acerca de la información del nombre de dominio que será enviada al servidor. Una vez que el DNS toma esta información da solución a estas preguntas; por el lado del servidor el BIND utiliza un demonio llamado *named*.

Después de que el servidor es habilitado, *named* es invocado cada vez que el sistema entra en modo multiusuario. Este lee información que encuentra en el archivo *named.boot* y toma las acciones apropiadas: prepara la cache, accesa las zonas de archivo y el *named* también es invocado de una línea de comando

El DNS tiene algunos términos claves que listan en la tabla 3.1.

TERMINO	DEFINICION
dominio.	Es la entidad lógica y la organización que representa una parte de una red.
Nombre de dominio.	Es la parte de un nombre de una computadora en una red, la cual representa el dominio que representa dicho nombre.
Anfitrión.	Es una computadora en una red.
Servidor de nombre.	Una computadora que proporciona servicios DNS para transformar nombres DNS en direcciones IP.
Resolvidor.	Es un programa o rutina de biblioteca que extrae información DNS de un servidor de nombre.
Resolución.	Es la acción de traducir un nombre DNS a su correspondiente dirección IP.
Resolución inversa.	Hacer concordar una dirección IP dada con su nombre DNS.

### **3.2. Configuración del resolvidor.**

Para usar el DNS se debe configurar como primer paso la biblioteca del resolvidor en la computadora personal. Se debe configurar el resolvidor local si se va a usar una resolución de nombres DNS, aunque no se vaya a ejecutar un servidor de nombre de dominio local..

El resolvidor no es un proceso distinto ni separado; es una rutina de biblioteca llamada por procesos de la red.

Hay dos formas de manejar la configuración del resolvidor. Se puede utilizar la configuración por default o crear una configuración típica.

La configuración por default :

- Usa el host local como un Servidor de Nombres por default.
- Deduce el nombre de dominio por default de una cadena que regresa un comando llamado `hostname`. Esto se hace para remover la parte de la cadena antes del primer punto, y se usa el resto de la cadena como el nombre de dominio.

#### **3.2.1 El archivo *host.conf*.**

El archivo *host.conf* se encuentra dentro del directorio `/etc` en el que se configuran las bibliotecas locales del resolvidor. Este archivo le indica al resolvidor cuáles servicios usar y en que orden, además lista opciones del resolvidor una por renglón. Para indicar el inicio de un comentario se utiliza el carácter especial `#`.

Para la configuración de este archivo se pueden utilizar diferentes opciones listadas en la tabla 3.2.

<b>Tabla 3.2 Opciones de configuración para el archivo /etc/host.conf</b>	
<b>OPCIÓN</b>	<b>FUNCIÓN</b>
<b>Order</b>	Indica en qué orden se desean ejecutar los diferentes opciones de resolución de nombres. Se ejecutan de acuerdo al orden listado en el archivo y tiene las siguientes opciones: <ul style="list-style-type: none"> <li>• <i>bind</i> Consulta un Servidor de Nombres DNS para resolver el nombre.</li> <li>• <i>hosts</i> Busca en el archivo local /etc/host y trata de resolver el nombre</li> <li>• <i>nis</i> Intenta resolver el nombre usando el protocolo de servicio de información de red (NIS)</li> </ul>
<b>Alert</b>	Tiene como argumentos <i>off</i> y <i>on</i> . Cuando se le especifica <i>on</i> , cualquier intento de engaño de una dirección IP se registra por medio de <i>syslog</i> .
<b>Multi</b>	Toma <i>off</i> y <i>on</i> como argumentos. Indica si a un anfitrión se le permite especificar más de una dirección IP en el archivo <i>etc/hosts</i> . <i>Multi</i> se usa solamente con consultas de anfitrión.
<b>Nospoof</b>	Se activa al especificar <i>nospoof on</i> . Cuando se utiliza la resolución inversa para hacer concordar un nombre de anfitrión con una dirección específica , se resuelve el nombre de anfitrión. Este se regresa para verificar que concuerde con la dirección que se consulta.
<b>trim</b>	Elimina el nombre de dominio antes de ejecutar una búsqueda de un nombre en <i>/etc/hosts</i> . Esto permite que se ponga el nombre base de anfitrión en <i>/etc/hosts</i> sin especificar el nombre de dominio.

Un ejemplo del archivo */etc/hosts* es el siguiente:

# Ejemplo del archivo */etc/hosts*

# Especifica el orden en va a buscar para resolver un nombre  
order bind host

#Permite máquinas con múltiples direcciones.  
multi off

# Alerta si existe un intento de engaño un una dirección IP  
alert on



En este ejemplo el resolvidor busca primero los nombres de anfitrión con DNS y luego lo intenta con el archivo local `/etc/hosts`. No se permite que haya direcciones IP múltiples para una sola máquina. Ajusta al resolvidor para que avise si hay un intento de hacer trampa (spoof)

### 3.2.2 El archivo `resolv.conf`.

El archivo `/etc/resolv.conf` controla la forma en como el *resolvidor* usa el DNS para resolver nombres de anfitrión. Establece los servidores de nombres DNS con los que hace contacto cuando resuelve un nombre de anfitrión y el orden en que lo hace. También proporciona el nombre de dominio local y algunas pistas para saber el nombre de algunos anfitriones, especificados sin este nombre.

Este archivo puede usar algunas de las opciones para su configuración listadas en la tabla 3.3.

<b>Tabla 3.3 Opciones de configuración para el archivo <code>/etc/resolv.conf</code></b>	
<b>OPCIÓN</b>	<b>FUNCIÓN</b>
Domain	Se utiliza para especificar el nombre de dominio local de este anfitrión. Si no se da, el resolvidor trata de obtener dicho nombre de dominio a partir de la llamada de sistema <code>getdomain name ()</code> .
Nameserver	Se utiliza para especificar la dirección IP del Servidor de Nombres, con el que hace contacto para la resolución de nombre se pueden especificar hasta tres direcciones IP de servidores y se van a acceder de acuerdo al orden presentado, es decir el primer servidor listado es el más confiable.
Search	Esta opción se utiliza cuando el nombre de dominio no es especificado como parte de una consulta de nombre de anfitrión, el <code>search</code> da una lista de dominios para probarla.

Ejemplo de la configuración del archivo *resolv.conf*.

```
domain nuts.com
```

```
#Primer servidor.
```

```
name server 127.0.0.1
```

```
#Siguiente servidor.
```

```
name server 128.66.12.1
```

```
#servidor final
```

```
name server 128.66.1.2
```

En este ejemplo, se especifica el dominio local por medio de la opción *domain* y aparece un nombre de servidor que se habrá de usar para resolver nombres de anfitrión.

En la opción *nameserver* se especifica la dirección IP del servidor de nombre y no el nombre de anfitrión ya que, si se especifica el nombre de anfitrión el DNS no sabe con que anfitrión debe establecer contacto.

### **3.3 Configuración de *named*.**

El demonio *named* al igual que otros demonios que utilizan servicios de Internet como el correo electrónico, gopher, ftp, etc. es un programa que se ejecuta en segundo plano se inicia casi siempre al momento de arranque y lee su información de configuración desde un conjunto de archivos de configuración. Una vez que el demonio arranca con la información de configuración escribe el ID de

proceso en archivo ASCII */etc/named.pit*. Luego comienza a escuchar las peticiones DNS en el puerto lógico de red, especificado por omisión en */etc/services*.

### **3.4 Desempeño del cache.**

El costo de una búsqueda para nombres no locales puede ser muy alto si se resuelve enviar hacia el servidor raíz. Incluso si las solicitudes pueden ir directamente hacia el servidor que tiene autoridad para el nombre, la búsqueda de nombres puede representar una carga para una red de redes.

Los servidores de nombres de Internet utilizan una memoria inmediata de los nombres (name caching) para optimizar las búsquedas. Cada servidor mantiene una memoria inmediata de los nombres utilizados más recientes, así como un registro de dónde fue obtenida la información para la asociación de nombres. Cuando un cliente interroga a un servidor a fin de resolver un nombre, el servidor verifica primero si tiene autoridad para el nombre de acuerdo con el procedimiento estándar. Si no es así, el servidor verifica su memoria inmediata para ver si el problema del nombre se resolvió recientemente. Los servidores reportan la información almacenada en memoria inmediata a los clientes, pero la marcan como una asignación no autorizada y entregan el nombre de dominio del servidor, desde el cual obtiene la asignación. El servidor local también envía información adicional que le indica al cliente la asignación entre el servidor y una dirección IP. De esta manera, los clientes reciben respuestas rápidamente, pero la información podría no estar actualizada. Si la eficiencia es importante, el cliente elegirá aceptar la respuesta no autorizada y continuar. Si la seguridad es importante, el cliente seleccionará contactar a la autoridad y verificar que la asignación entre el nombre y la dirección siga siendo válida.

El procedimiento mediante el uso de memoria inmediata trabaja bien en el sistema de nombres de dominio debido a que las asignaciones de nombres a direcciones cambian con poca frecuencia. Sin embargo, éstas se modifican. Si el servidor capturó la información la primera vez que le fue solicitada y nunca la cambió, las entradas de información de la memoria inmediata podrían estar incorrectas. Para mantener la información correcta, los servidores cronometran cada entrada y suprimen las entradas que excedan un tiempo razonable. Los servidores no aplican un sólo límite de tiempo fijo para todas las entradas, pero permiten a la autoridad de una entrada configurar su límite de tiempo. Cada vez que una autoridad responde a una solicitud, incluye un valor de Tiempo de Vida (time to live ó ttl) en la respuesta, en la que especifica que tanto se garantiza la conservación de la asignación.

### 3.5 El archivo *named.boot*.

El primer archivo que ejecuta *named* cuando inicia es el archivo */etc/named.boot* este es el que enlaza todos los demás archivos de configuración que usa *named*. Contiene apuntadores a los diversos archivos de configuración y a otros servidores de nombre. En este archivo los comentarios se inician con ; . Cuenta con varias opciones que son las mas usuales y se listan en la tabla 3.4.

Tabla 3.4 Opciones de configuración para el archivo <i>named.boot</i>	
OPCIÓN	FUNCIÓN
Directory	Es el directorio donde están ubicados los archivos de zona DNS . Se pueden especificar diferentes directorios utilizando varias veces <i>directory</i> .
Primary	Esta opción toma un nombre de dominio y uno de archivo como argumentos. <i>Primary</i> declara a <i>named</i> confiable para el dominio específico y le hace cargar la información de zona desde el archivo especificado.

Secondary	Al usar esta opción <i>named</i> actúa como servidor secundario para el dominio específico. Toma un nombre de dominio una lista de direcciones y un nombre de archivo como argumentos. <i>Named</i> trata de transferir la información de zona de los anfitriones especificados en la lista de direcciones y luego guarda la misma información en el archivo especificado en la línea de opción.
Cache	Establece la información en la memoria cache para <i>named</i> . Utiliza un nombre de dominio uno de archivo como argumentos, el primero se especifica como un punto (.). El archivo contiene un conjunto de registros conocidos como pista de servidor, donde se lista información acerca de los servidores de nombre de raíz.
Forwarders	Toma una lista de servidores de nombre como argumentos. Le informa al Servidor de Nombres local que trate de establecer contacto con los servidores de la lista que tomó, en caso de no poder resolver una dirección a partir de su información local.
Slave	Se utiliza para convertir el Servidor de Nombres local en un servidor esclavo.

Ejemplo del archivo *named.boot*

; nuts.com servidor primario en el archivo *name.boot*

```

directory                /var/named
primary nuts.com          named.hosts
primary 66.128.IN-ADDR.ARPA  named.rev
primary 0.0.127.IN-ADDR.ARPA  named.local
cache                      named.ca

```

En este ejemplo, la opción *directory* le indica a *named* que todos los archivos de trabajo están en el directorio */var/named*. El servidor principal de nombre es *nuts.com*, y los registros de información de zona de anfitrión están en el archivo *named.hosts*. Hay una segunda línea *primary* que muestra que también se tiene

autoridad de zona primaria para la zona 66.128.in-addr.arpa, con información de zona en el archivo *named.rev*. La tercera línea indica que se tiene autoridad sobre la zona primaria del localhost y se especifica en *named.local*. Por último la línea de cache establece la información almacenada en la memoria caché para este servidor de nombre. Esta opción debe estar en casi cualquier máquina que se ejecute como servidor de nombre. Le dice a *named* que permita la caché y cargue la información de servidor de raíz desde el archivo *named.cs*.

La sintaxis IN-ADDR.ARPA es la forma de *named* para hacer concordar las direcciones IP con los nombres DNS. Como el DNS se estableció en su origen para hacer concordar nombres DNS con direcciones, necesita una línea diferente *primary* para realizar la resolución inversa.

### **3.6 Archivos de bases de datos y registro de recurso.**

Los archivos usados en la configuración *named* como son : *named.host*, *named.rev* , *named.local* y *named.cs* almacenan información del dominio en una base de datos. Todos estos archivos tienen el mismo formato básico y usan el mismo tipo de registro para definir información del dominio en la base de datos. Los registros que estos usan son registros de recursos estándar, llamados RRs.

Los registros de recursos usan una sintaxis general . Sin embargo varias partes del registro son opcionales, conforma el tipo de registro, y pueden asumir un valor por omisión cuando no se especifican. El formato de un registro de recurso es:

{propietario} {ttl} clase tipo de registro tipo de dato

**Propietario.** El campo propietario se refiere al nombre de dominio o anfitrión al que se aplica el registro.

**Ttl.** El ttl es el campo tiempo de vida (time to live) especifica en cuanto tiempo en segundos se valida la información de este registro después de recuperarla de un servidor DNS.

**Clase.** Clase se especifica una clase de dirección de red. Para redes TCP/IP se usa el valor in.

**Tipo de registro.** Este campo muestra el tipo de este registro

**Tipo de dato.** El formato de campo de datos depende del contenido del campo de tipo de registro.

En la tabla 3.5 se listan los tipos de registros de recurso más comunes

<b>Tabla 3.5 Tipos de registro de recursos</b>	
<b>TIPO</b>	<b>DESCRIPCION</b>
<b>A</b>	Es un registro de dirección. Asocia un nombre de anfitrión con una dirección. Únicamente puede existir un solo registro A para cualquier anfitrión dado.
<b>CNAME</b>	Asocia un alias para un anfitrión con el nombre especificado en el registro A para este anfitrión.
<b>HINFO</b>	Proporciona información acerca de un anfitrión
<b>MX</b>	Establece un registro que intercambia correo. Los registros MX le ordenan al transporte de correo que envíe el correo a otro sistema que sepa la manera de entregarlo a su destino final
<b>NS</b>	Apunta a un servidor de nombre de otra zona. El campo de dato de este registro contiene el nombre DNS del Servidor de Nombres. Se requiere especificar también un registro A para que corresponda con el nombre de anfitrión que tiene la dirección del servidor de nombre
<b>PTR</b>	Convierte direcciones a nombres de hosts

**Tabla 3.8 Continuación**

SOA	<p>Registro de inicio de autoridad. Marca el comienzo de una zona de datos y define parámetros que afecten la zona de entrada. El SOA contiene los siguientes datos:</p> <ul style="list-style-type: none"><li>• <b>origin</b> Es el nombre de anfitrión del servidor de nombre primario para este dominio. Se da como un nombre de dominio absoluto y termina con un punto (.), por lo que no lo modifica <i>named</i>.</li><li>• <b>Contact</b>. Es el contacto de correo electrónico de la persona responsable de mantener este dominio. Se utiliza el carácter de punto (.) para determinar una dirección de correo electrónico.(Continua)</li><li>• <b>Serial</b>. Es el número de versión del archivo de información de zona, que se da como un entero. Lo usan los DNS secundarios para especificar cuándo ha cambiado el archivo de información de zona. Se incrementa en una en cada modificación.</li><li>• <b>Refresh</b>. Es la cantidad de tiempo en segundos que debe esperar un servidor secundario, antes de intentar revisar el registro SOA del servidor de nombre primario.</li><li>• <b>retry</b>. Es el tiempo en segundos que un servidor secundario espera para volver a intentar una petición al servidor primario, si el servidor primario no está disponible.</li><li>• <b>Expire</b>. Es el tiempo en segundos que el servidor secundario debe esperar, antes de desechar la información de zona, si no pudo hacer contacto con el servidor primario.</li><li>• <b>Minimum</b>. Es el valor ttl por omisión para los registros del recurso que no lo especifica.</li></ul>
-----	---

### 3.7 El archivo *named.local*.

El archivo *named.local* es usado para convertir la dirección 127.0.0.1 (dirección local) dentro de el nombre localhost. Esta es la zona de archivo para el dominio



inverso 0.0.127.IN.-ADDR.ARPA. Porque todos los sistemas usan 127.0.0.1 como la dirección local, este archivo es idéntico en cada servidor. El siguiente ejemplo es una configuración típica del archivo *named.local*:

```
@ IN SOA almond.nuts.com. jan.almond.nuts.com. (
    1 ; serial
    36000 ; refresco cada 100 horas
    3600 ; reintento después de una hora
    3600000 ; expira después de 1000 horas
    36000 ; mínimo 100 horas
)
IN NS almond.nuts.com.
1 IN PRT localhost.
```

Los datos de los registros en los campos SOA y NS indican el nombre del Servidor de Nombres, que varía de sistema a sistema. En el ejemplo el registro SOA identifica a almond.nuts.com. como el servidor originario de esta zona, y la dirección de correo jan.almond.nuts.com. como el punto de contacto para algunas cuestiones acerca de esta zona. Usualmente en la dirección de correo electrónico se utiliza la @ ,pero en este caso es cambiada por un punto. Después se listan algunos datos que debe contener un archivo SOA y por último la línea que indica el registro PRT que es la que realiza la resolución inversa en localhost

### **3.8 El archivo *named.hosts*.**

El archivo *named.hosts* contiene la mayoría de la información de el dominio local. Este archivo convierte nombres de anfitrión a direcciones IP. El archivo *named.hosts*, como el archivo *named.rev*, son creados únicamente en el servidor primario. Todos los demás servidores obtienen esta información de el servidor primario.

A continuación se lista un ejemplo del archivo *named.hosts*:

```
;Direcciones y otra información del host.
;
@      IN      SOA      almond.nuts.com  jan.almond.nuts.com (
                                10118 ; serial
                                43200 ; refresh
                                3600   ; retry
                                3600000 ; expire
                                2592000) ; minimum
;
                                IN      NS      almond.nuts.com.
@      IN      A        128.66.12.1
                                IN      MX      10      almond.nuts.com.
                                IN      HINFO    PC-486   Linux
;
;      Define el localhost

localhost      IN      A        127.0.0.1
```

La primera línea que se encuentra en este archivo es el SOA o inicio de autoridad para nuestro dominio. La primera línea de este registro comienza con el carácter @, que indica el origen actual o dominio almond.nuts.com. La definición del origen viene del dominio listado en la línea primary correspondiente en *named.boot*. Los códigos IN SOA le indican a *named* que este registro de recurso usa direccionamiento Internet (TCP/IP) y es un registro de inicio de autoridad. Los siguientes argumentos son el nombre canónico del servidor de nombre primario para este dominio y el contacto de correo electrónico con el carácter @ reemplazado por un punto jan.almond.nuts.com. La siguiente línea lista a almond.nuts.com como

servidor de nombre como el dominio propio. A la siguiente línea se le asigna la dirección IP de almond.nuts.com. La siguiente línea pone un registro intercambiador de correo MX para nuts.com . La última línea de este grupo establece un registro de información de anfitrión HINFO, el cual indica que es una PC 486 y que ejecuta Linux. Finalmente se especifica el localhost que esta asignado a la dirección 127.0.0.1.

### 3.9 El archivo *named.rev*.

El archivo es muy similar en estructura al *named.hosts*, con la diferencia de que trabaja a la inversa. Asigna direcciones a nombres de anfitrión. Ambos archivos cuentan con registros PTR.

```
@      IN      SOA  almond.nuts.com  jan.almond.nuts.com (
                                10118 ; serial
                                43200 ;refresh
                                3600  ;retry
                                3600000 ;expire
                                2592000) ; minimum
;
                                IN      NS      almond.nuts.com.

1      IN      PTR   almond.nuts.com
2      IN      PTR   mailhost.nuts.com
3      IN      PTR   opus.nuts.com
```

En este ejemplo lo único que hay que especificar son los registros PTR, los cuales conforman los registros de resolución de dirección inversa y dan el número de anfitrión (la parte de la dirección IP sin listar en el valor in-addr-arpa del *named.boot*), además del nombre de anfitrión canónico que le corresponde.

### 3.10 El archivo *named.ca*

El archivo *named.ca* establece la cache, es el más simple de los archivos de configuración *named*, pero su operación es muy importante. Incluye los servidores de nombre de raíz para los diversos dominios, junto con sus direcciones IP. Contiene también indicadores de campo especiales que le dicen a *named* que estos son servidores de raíz.

El siguiente formato es un ejemplo del *named.ca*:

```
;archivo named.ca
```

```
99999999 IN NS NS.NIC.DDN.MIL.  
99999999 IN NS NS.NASA.GOV.  
99999999 IN NS KAVA.NISC.SRI.COM.  
99999999 IN NS TERP.UMD.EDU.  
99999999 IN NS C.NYSER.NET.  
99999999 IN NS NS.INTERNIC.NET.
```

```
NS.NIC.DDN.MIL. 99999999 IN A 192.112.36.4  
NS.NASA.GOV. 99999999 IN A 128.102.16.10  
KAVA.NISC.SRI.COM 99999999 IN A 192.33.33.24  
TERP.UMD.EDU. 99999999 IN A 128.8.10.90  
C.NYSER.NET. 99999999 IN A 192.33.4.12  
NS.INTERNIC.NET 99999999 IN A 198.41.0.4
```

El archivo *named.ca* sólo asigna los registros de nombre de servidor NS a las direcciones IP adecuadas para ellos.

Para obtener una lista actual completa de los servidores de nombre de raíz, se utiliza nslookup.

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**

**CAPITULO 4**  
**CASO PRACTICO**

## **4 CASO PRÁCTICO. CONFIGURACIÓN DE UN SERVIDOR DE NOMBRES (DNS) DENTRO DE LA FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN.**

### **4.1 Justificación de la instalación.**

La instalación del Servidor de Nombres se llevó a cabo en el centro de cómputo de la Facultad de Estudios Superiores Cuautitlán, ya que no se contaba con un Servidor de Nombres propio para el acceso a redes externas como la red Internet, esto hacía que los accesos fueran más lentos, porque tenían que conectarse al Servidor de Nombres de DGSCA ubicado en las instalaciones de la Ciudad Universitaria.

Debido a las razones mencionadas, se planeó la instalación del Servidor de Nombres en la Facultad bajo el sistema operativo Linux, como ya se explicó no requiere de licencias y no tiene ningún costo su uso, además de que cuenta con características flexibles de red y facilita la configuración del Servidor de Nombres de una forma gratuita.

Los beneficios que nos dará la instalación del servidor son:

- Disminuye la carga de las solicitudes para la resolución de nombres.
- Agilizar los accesos a las redes externas dentro de la Facultad de Estudios Superiores Cuautitlán.

### **4.2 Consideraciones Administrativas.**

Antes de que una institución obtenga autorización oficial para un dominio de segundo nivel, debe de estar de acuerdo en operar un Servidor de Nombres de

dominio que cumpla con los estándares de Internet. El Servidor de Nombres de dominio debe de obedecer el estándar de protocolo que especifica los formatos de mensaje y las reglas para responder a las solicitudes. El servidor también debe conocer las direcciones de los servidores que maneja cada subdominio (si existe uno) así como la dirección del último servidor de raíz.

La implantación de servidores también es complicada pues la autoridad Internet requiere que la información en todos los servidores de dominio sea respaldada. La información debe aparecer en por lo menos dos servidores que no operen en la misma computadora. Los servidores no deben tener punto común alguno de falla. Evitar los puntos comunes de falla significa que dos servidores de nombre no pueden estar conectados a la misma red; éstos no pueden obtener suministro eléctrico de la misma fuente. Así, para cumplir con los requerimientos una localidad debe encontrar al menos otra localidad con la que acuerde operar un respaldo del Servidor de Nombres. En cualquier punto del árbol de servidores, un servidor debe saber como localizar los servidores de nombres primarios y de respaldo para los subdominios y debe dirigir sus solicitudes hacia un Servidor de Nombres de respaldo si el servidor primario no está disponible.

Con respecto al caso práctico la instalación se llevó a cabo en el subdominio `cuautitlan2.unam.mx`, por lo que se debe de dar de alta en un servidor con una jerarquía mayor como son los servidores de RedUNAM. El registro lo lleva a cabo el administrador del dominio `cuautitlan2.unam.mx`, llenando un formato de registro de DNS que emite la administración de DNS de DGSCA con la siguiente información:

Debido al crecimiento de la RedUNAM y a la necesidad de utilizar el Servidor de Nombres (DNS), para llevar un control de las altas, bajas y modificaciones que se llevan a cabo dentro de este servicio, se deben de seguir ciertas políticas para dar la atención mas apropiada y eficiente.



1. La solicitud de altas, bajas o modificaciones la hará el responsable de la red.

2. La solicitud del servicio tendrá el siguiente formato:

DEPENDENCIA :  
RESPONSABLE :  
CORREO ELECTRONICO :  
CARGO :  
TELEFONO :  
DOMINIO :  
NOMBRE DE LA MAQUINA:  
DIRECCION IP :  
MARCA DE LA MAQUINA :  
MODELO DE LA MAQUINA:  
SISTEMA OPERATIVO :  
VERSION DEL S.O. :  
TIENE SNMP ?

Cada uno de los campos son de útil importancia para llevar la Administración de los equipos con que se esta trabajando.

Donde:

**DEPENDENCIA** es la Institución que representa cada red.

**RESPONSABLE** es la persona que maneja y administra a las máquinas de su dominio (segmento de red). En el caso de que sean muchas máquinas, únicamente bastara con que se escriba una sola vez.

**CORREO ELECTRONICO** es el campo que requiere la colocación de la dirección de correo electrónico del responsable.

**CARGO** es la ocupación que tiene el responsable dentro de la dependencia.

**TELEFONO** Se colocara el numero telefónico y extensión correspondiente.

**DOMINIO** Es el dominio de red al cual le corresponde la máquina.

Ejemplo: Para la máquina : xel-ha.cuautitlan2.unam.mx

Su dominio es : cuautitlan2.unam.mx

**NOMBRE DE LA MAQUINA** Será el nombre que el responsable asigne a la máquina, este nombre no excederá de 10 caracteres, de lo contrario, si se excede este número se cortara la cadena a 10 caracteres.

**DIRECCION IP** Será la dirección Internet. Ej. (132.248.102.21)

**MARCA DE LA MAQUINA** Se refiere a la compañía de fabricación del equipo.

**MODELO DE LA MAQUINA** Se refiere a las especificaciones del equipo.

**SISTEMA OPERATIVO** El tipo de sistema operativo que utilizan los equipos.

**VERSION DEL S.O** Es la versión actual del sistema operativo.

**SNMP** Esta opción es opcional, y únicamente los que tengan este protocolo.

**Nota :** Es importante recalcar el punto sobre el **NOMBRE DE LA MAQUINA**.

**3. La solicitud estará ordenada en orden descendente conforme la dirección IP.**

4. La atención a las solicitudes se harán conforme vayan llegando.

5. La solicitud será enviada vía correo a la cuenta de :

`dns@noc.noc.unam.mx`

Siguiendo con las anteriores políticas se asegura una rápida atención a las solicitudes presentadas. De lo anterior se pretende entablar una comunicación entre los responsables de red con la administración del Servidor de Nombres.

#### **4.3 Instalación del Servidor de Nombres (DNS).**

Se considera que Linux esta previamente instalado con sus características básicas y de red. Posteriormente se deben de verificar la existencia de algunos archivos como son:

el demonio `/usr/sbin/named`

Este es el archivo más importante para la instalación del Servidor de Nombres, el cual se debe de cargar dentro de un directorio `etc/rc`, ya que en este directorio es donde se ejecutan los archivos principales al arrancar el sistema. Por esta razón se establece la siguiente línea al final del archivo `/etc/rc.d/rc.local`:

```
/usr/sbin/named -b /etc/named.boot
```

Si no se cuenta con este archivo se puede conseguir en la siguiente dirección:  
<http://www.redhat.com>.

Los archivos que deben de existir son:

```
/etc/named.boot  
/var/named/named.hosts  
/var/named/named.ca  
/var/named/named.rev  
/var/named/named.local
```

En caso de que no existan se deben de crear en las rutas especificadas utilizando algún editor de Linux.

#### **4.4 Configuración de archivos.**

El primer paso es configurar el archivo *named.boot*, en donde existen los archivos maestros:

directory		<i>/var/named</i>
cache		<i>named.ca</i>
primary	<i>cuautitlan2.unam.mx</i>	<i>named.hosts</i>
primary	<i>102.248.132.IN-ADDR.ARPA</i>	<i>named.rev</i>
primary	<i>0.0.127.IN-ADDR.ARPA</i>	<i>named.local</i>

Una vez hecho lo anterior se deben de configurar los archivos maestros. Comenzaremos con la configuración de *named.hosts*:

```
var/named/named.hosts
```

```
;/var/named/named.hosts. Hosts local en cuautitlan2.unam.mx
```

```
; Origen cuautitlan2.unam.mx
```

```

@ IN SOA xel-ha.cuautitlan2.unam.mx hduarte.cuautitlan2.unam.mx
(
    1 ; serial
    86400 ;refresco: una vez por día
    3600 ;reintento: una hora
    3600000 ;expira: 42 días
    604800) ; mínimo: una semana
;
    IN NS xel-ha.cuautitlan2.unam.mx

; Nombre de dominio propio

@ IN A 132.248.102.21
    IN MX 100 xel-ha.cuautitlan2.unam.mx
    IN HINFO PC-586 Linux
;
; Dirección local

localhost IN A 127.0.0.1

;servidor fesc
fesc IN A 132.248.102.36

;otros equipos
www IN A 132.248.102.34
irxcicc IN A 132.248.102.33
cicc IN A 132.248.102.30

```

El siguiente archivo es */var/named/named.local*

```
;/var/named/named.local
```

```
@ IN SOA xel-ha.cuautitlan2.unam.mx.
```

```
hduarte.cuautitlan2.unam.mx. (
```

```
1 ; serial  
86400 ;refresco: una vez por día  
3600 ;reintento: una hora  
3600000 ;expira: 42 días  
604800) ; mínimo: una semana
```

```
;
```

```
IN NS xel-ha.cuautitlan2.unam.mx
```

```
1 IN PTR localhost.
```

La siguiente configuración es referente al archivo */var/named/named.rev*

```
;/Direcciones y otra información del host.
```

```
;
```

```
@ IN SOA xel-ha.cuautitlan2.unam.mx hduarte.cuautitlan2.unam.mx
```

```
(
```

```
1 ; serial  
86400 ;refresco: una vez por día  
3600 ;reintento: una hora  
3600000 ;expira: 42 días  
604800) ; mínimo: una semana
```

```
;
```

```
IN NS xel-ha.cuautitlan2.unam.mx
```

```
1 IN PTR xel-ha.cuautitlan2.unam.mx
```

2	IN	PTR	fesc.cuatitlan2.unam.mx
3	IN	PTR	irixcicc.cuautitlan2.unam.mx
4	IN	PTR	www.cuautitlan2.unam.mx

Finalmente se configura el archivo *var/named/named.ca*

;archivo named.ca

;

```

99999999 IN NS NS.NIC.DDN.MIL.
99999999 IN NS NS.NASA.GOV.
99999999 IN NS KAVA.NISC.SRI.COM
99999999 IN NS TERP.UMD.EDU.
99999999 IN NS C.NYSER.NET.
99999999 IN NS NS.INTERNIC.NET.

```

;

```

NS.NIC.DDN.MIL.      99999999 IN A 192.112.36.4
NS.NASA.GOV.        99999999 IN A 128.102.16.10
KAVA.NISC.SRI.COM   99999999 IN A 192.33.33.24
TERP.UMD.EDU.       99999999 IN A 128.8.10.90
C.NYSER.NET.        99999999 IN A 192.33.4.12
NS.INTERNIC.NET.    99999999 IN A 198.41.0.4

```

Después de haber configurado los archivos *maestros*, hay que verificar el orden en que va a resolver el archivo */etc/host.conf*. El formato que debe de tener es el siguiente:

```

order bind hosts
multi off
spooof on
alert on

```

Hecho esto se prosigue a especificar la dirección del Servidor de Nombres que acabamos de crear en el archivo *resolv.conf*, con el siguiente formato:

```
domain cuautitlan2.unam.mx
```

```
nameserver 132.248.102.21
```

Otro punto interesante a verificar es la igualdad de direcciones IP del Servidor de Nombres que creamos y del hostlocal, la cuales deben de coincidir con las direcciones IP establecidas en el archivo */etc/hosts*

Por último estamos en condiciones de probar nuestro Servidor de Nombres de dominio, primero reiniciar el equipo y posteriormente utilizar *nslookup* el cual consulta el DNS y regresa el nombre de Servidor de Nombres actual y su dirección IP. Estos datos deben ser los que especificamos en la configuración del servidor:

Nombre: *xel-ha.cuautitlan2.unam.mx*

Dirección: *132.248.102.21*

Una vez que regrese estos valores podemos probar el acceso a otras direcciones de DNS, por ejemplo: *www.unam.mx*

#### **4.5 Solución de problemas**

El DNS es un sistema muy complejo. Muchos de los problemas que suceden en la configuración de un DNS son el resultado de errores de sintaxis en los archivos de configuración.



Por esta razón hay que cerciorarse de que los nombres de anfitrión en los archivos de configuración DNS estén escritos de manera correcta. Si se trata de un nombre de anfitrión absoluto, hay que asegurarse de que termine con un punto.

Otro punto que hay que tomar en cuenta es tener cuidado con los nombres usados en los registros SOA y CNAME si se cometen errores en esta parte, estos registros de recurso pueden dirigir de nuevo las consultas hacia nombres de anfitrión de computadoras que no existen.

#### **4.6 Administración del Servidor de Nombres.**

Cada sistema debe de tener su propio administrador o persona encargada de que todo éste a punto en todo momento, los usuarios confían en que los administradores garantizarán la seguridad e integridad del sistema.

A continuación se listaran algunas de las tareas del administrador :

- Planificación e instalación.
- Planificación para el crecimiento futuro
- Administración de usuarios.
- Mantener la seguridad del software y los datos.
- Mantener la seguridad del equipo.
- Mantener el rendimiento

Además es recomendable que dentro del archivo *host.conf* se establezcan la opciones *spoof* y *alert* para evitar que existan engaños en las direcciones IP.

Otra recomendación es que el Servidor de Nombres se encuentre conectado a un UPS (fuente de poder de energía ininterrumpible) para no privar a los usuarios de este servicio y evitar la caída del servidor.

## **5 CONCLUSIONES**

Con la utilización de Linux en la instalación y configuración del Servidor de Nombres se determinó que es un sistema operativo capaz, flexible y confiable para realizar tareas de red. Además con su característica de ser gratuito y debido a que no se requiere un equipo muy sofisticado para su instalación se disminuyen los costos al ser usado en pequeñas y grandes redes.

La configuración del DNS resulta un poco complicada, porque es difícil comprender los parámetros y se debe poner atención a la sintaxis, ya que la falta de un carácter causa problemas en el funcionamiento. También hay que tener especial atención en la lógica y estructura en que se relacionan los archivos, pero una vez configurado su mantenimiento y administración es fácil.

Se logró la instalación del Servidor de Nombres dentro de la Facultad de Estudios Superiores Cuautitlán, con lo que la Facultad cuenta con un Servidor de Nombres propio y de esta manera se disminuye la carga de la red local, ya que cuando se solicite resolver un nombre no es necesario hacer una conexión a redes externas puesto que el Servidor de Nombres consultará su memoria inmediata o su base de datos y así podrá resolver la solicitud.

A diferencia de otros sistemas operativos, la instalación de un Servidor de Nombres con Linux es más confiable y segura, debido a que cuenta con características de Unix, el cual viene manejando desde hace tiempo este tipo de servicios.

## BIBLIOGRAFIA

1. Albitz, Paul. DNS and BIND 2ª edición. E.U.: O'really, 1996. 438 p.
2. Tackett Jack, Jr. Linux. México: Prentice Hall, 1996. 831p.
3. Welsh, Matt. Running LINUX 2ª edición. E.U.: O'really, 1996. 650 p.
4. E. Comer Douglas. Redes Globales de Información con Internet y TCP/IP. 3ª edición. México: Prentice Hall, 1996. 607 p.
5. Dowd, Kevin. Getting Connected. E.U.: O'really, 1996. 424 p.
6. LeBlanc Dee-Ann. Construya un site perfecto en Internet con Linux. México: Prentice Hall, 1996. 379 p.
7. Cricked, Liu. Managing Internet Information Service 4ª edición. E.U.: O'really, 1994. 668 p.
8. Feit, Sidnie. TCP/IP: architecture, protocols and implementation. New York: McGraw Hill. ,1993. 466p (Computer communications).
9. Hunt, Craig. TCP/IP Network Administration. Sebastopol California: O'Reilly, 1994. 472 p
10. Frisch, Aileen. Essential System Administration 2ª edición. E.U.: O'really, 1995. 788p
11. <http://sunsite.unc.edu/mdw/LDP/nag>.