UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE AUTOMATIZACIÓN DEL DNS (DOMAIN NAME SYSTEM)

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

ARMANDO ANTONIO AGUILAR ALVAREZ

DIRECTOR DE TESIS:
ING. SERGIO NOBLE CAMARGO

MÉXICO, D.F. AGOSTO DE 1996

TESIS

COMPLETA

AGRADECIMIENTOS

A Dios:
Doy gracias, por haberme concedido la vida y por guiar siempre mi camino.

A mis Padres Armando Aguilar y Luz Amparo Alvarez:
A ustedes mis mejores amigos y los mejores maestros de toda la vida, que con su gran amor y cariño confiaron siempre en mí. Les doy gracias por haberme enseñado y mostrado el ejemplo de unión, lucha, entrega y dedicación por la vida y el trabajo.

A mis Hermanas Dolores de Jesús, Amparo de los Angeles y Ma. Eugenia:
Por su ánimo, valor y paciencia, así como su ejemplo, amor, cariño y confianza.

A mi Querida Esposa Rocío:
Por renacer las ilusiones y esperanzas en mí y porque formemos una historia común con dirección siempre hacia arriba, juntos con los ojos abiertos hacia lo que buscamos profundamente, como en los sueños del despierto que quiere tocar lo infinito y lo consigue, gracias por tu gran amor, ternura y cariño.

A mis Amigos:
Mauricio, créeme que valió la pena todo el esfuerzo personal y profesional. Gracias a todos tus consejos he alcanzado parte de mi realización como persona.
Antonio, por aquellas noches interminables de trabajo y por compartir momentos difíciles en nuestras vidas.
Sergio H., por conservar nuestra amistad y por confiar siempre en mi.
Paty, por apoyarme en todo momento.
Mike, por ser mi amigo, compañero y maestro.
Sergio N., gracias por tus consejos y por brindarme tu amistad.

A los Ingenieros:
Sergio Noble Camargo y Michael de Leo Gayol por su enseñanza y apoyo.

A Soluciones Avanzadas de Redes S.A. de C.V.:
Por brindarme las oportunidades y condiciones necesarias para la terminación de este trabajo. Así como mi profundo agradecimiento al personal que ahí labora.

A la Facultad de Ingeniería:
Por ser mi cuna profesional y por permitir que mi esfuerzo constituya parte de su historia. También por forjar en mí el espíritu universitario y seguir siempre la tendencia de que la clase universitaria es la que logrará la transformación del país.

ÍNDICE

LISTA DE FIGURAS

Lista de Tablas

INTRODUCCION

En las redes de computadoras, lo principal es el compartir recursos. La información, piedra angular de la computación, permite a grupos de personas organizarse para lograr el intercambio de información a grandes distancias, integrando al mundo con tecnología. Básicamente, tener la forma de accesar datos, transferirlos y permanecer bajo el concepto de estar en una red, permite que el mundo esté interconectado.

Internet, la red mundial más grande de información, es conocida como la red de redes, se integra al mundo bajo su estructura primaria de mantener comunicadas organizaciones de diferentes áreas como son: Educación, Gobierno, Militar, Comercial y organizaciones privadas. En la actualidad, Internet es una de las redes más populares ya que debido a su versatilidad proporciona a los usuarios grandes beneficios. Es una herramienta para la explotación de los grandes bancos de información que hoy en día están disponibles para el mundo entero.

Existen dentro de Internet una gran variedad de aplicaciones que permiten realizar búsquedas y presentación de información. Continuamente la red Internet crece en forma desmedida. El interés principal para conectarse a ella es que además de ser una red de computadoras, es una red de servicios y recursos; es una biblioteca, una base de datos y una comunidad de personas de todos los estilos de vida, listos para responder preguntas, escuchar y compartir. Es una razón electrónica verdaderamente social.

A medida que las aplicaciones de la Internet se fueron haciendo más sofisticadas, se vió en la necesidad de realizar un esquema general de servicio de nombres para reconocer a los servidores de información. En un principio se concibieron algunas formas: La tabla de *hosts* (computadoras huéspedes) del Network Information Center (NIC, Centro de Información de la Red), la tabla de *hosts* de cada servidor, el Network Information System (NIS, Sistema de Información de la Red) y el Domain Name System (DNS, Sistema de Nombre de Dominio), este último es el mejor mecanismo de proporcionar un servicio de resolución de nombres.

El DNS es un sistema jerárquico formado por bases de datos que están distribuidas por la red y tienen disponibilidad de proporcionar información relacionada a los *hosts* que pertenecen a una red, traducen de nombre a dirección; este sistema se puede administrar localmente, lo que permite agregar, quitar o cambiar información de un *host* en particular y a través de este manejo, todas las

redes conectadas entre sí comparten información distribuida sin centralizar el flujo de datos.

La administración local en un sistema de forma distribuida es de gran importancia, ya que evita que todo *host* que pertenezca a alguna red sea registrado en el NIC centralizadamente como antes se hacía, ahora en lugar de eso, cada ***servidor de nombres*** tiene que ser registrado ahí.

La consistencia, robustez y buena administración del DNS permite un mejor funcionamiento del uso de las aplicaciones de cualquier red, es decir, la adecuada operación de este sistema/servicio garantiza que las aplicaciones sean confiables. De ahí la importancia del DNS dentro de la configuración de cualquier nodo de Internet.

El DNS se ha convertido en un servicio básico de configuración en cada red basada en *Transmission Control Protocol/Internet Protocol* (*TCP/IP*, que más adelante se comentará).

Existen algunas razones importantes para trabajar con el DNS, tales como son intercambiar correo electrónico de un tipo de red a otra, reconocer a los *hosts* de otras redes sin tener la tablas de *hosts* de todos los servidores que las conforman, mejor repuesta en la conectividad, confiabilidad entre conexiones, entre otras aplicaciones.

Es de gran relevancia tener este servicio configurado en uno o más servidores de nuestra red, debido a que se tiene la ventaja de poder accesar a aplicaciones mediante el nombre de un *host*.

El DNS aporta una gran simplificación en la utilización de un nombre, en lugar de un número para identificar al *host*.

El propósito de este trabajo es dar a conocer a la comunidad de las redes de datos, la importancia que tiene la utilización del ***Sistema de Nombre de Dominio (DNS)*** como parte de la configuración básica de una máquina dentro de Internet, por lo que, se pretende proporcionar una herramienta que logre optimizar la administración de este servicio.

El origen de este trabajo experimental, es proporcionar un método en la configuración, mantenimiento y administración del DNS, un procedimiento y guía para hacer eficiente el servicio.

Se pretende que esta tesis describa aspectos generales del Sistema de Nombre de Dominio (DNS), para luego dar a conocer las facilidades del Sistema

de Automatización que se ha desarrollado. De modo que se ha dividido de la forma siguiente:

En el Capítulo 1, se presentan los fundamentos y antecedentes del DNS, es decir, la historia, los conceptos básicos de los servidores de nombres y sintaxis de los tipos de registros que se manejan.

En el Capítulo 2, se aprecia la Metodología utilizada en el trabajo. En este capítulo se desglosa la técnica utilizada para realizar el sistema basado propiamente en el concepto de Ingeniería de Software.

De modo que en el Capítulo 3, se realizará el Análisis y Diseño del SADNS (Sistema de Automatización del DNS) donde se tiene un papel importante ya que en este tema se plasman los acontecimientos y eventos que ocurren para que este proyecto se realizara. Se presentan los pasos siguiendo con la metodología utilizada.

En el Capítulo 4, se describe el Desarrollo del Sistema. Se revisa la utilización de los recursos para llevar a cabo el sistema, así como las herramientas y lenguajes de programación utilizados para desarrollar el sistema.

Finalmente en el Capítulo 5, se presentan las Conclusiones de la tesis realizada, que permitan conjuntar las ideas centrales del trabajo expuesto.

OBJETIVOS

- Optimizar el Sistema de Nombre de Dominio a través de procedimientos para la actualización automatizada de la información que proviene de cada red conectada a la RedUNAM.

- Dar a conocer a los usuarios de la RedUNAM la importancia que tiene el Sistema de Nombre de Dominio dentro de los servicios de la red Internet.

- Aprovechar los recursos de Internet para hacer interfaces de acceso remoto, así como administrar servicios de red fácilmente.

1. ANTECEDENTES

1.1 HISTORIA DE INTERNET

Internet surgió a consecuencia de que la Defense Advanced Research Project Agency (DARPA, Agencia de Proyectos Avanzados de Investigación del Departamento de Defensa de E.U.A.) fundó una red de computadoras de tipo experimental de área amplia que cubrió a los Estados Unidos, a la que se le llamó ARPANET. La meta fundamental de este proyecto fue permitir a militares compartir recursos de cómputo y estrategias de guerra. Otro de sus objetivos fue el diseñar una red que no se dañara con facilidad al perder uno o algún segmento físico, es decir, se deseaba que la red permitiera sumar nodos y modificar su estructura cuando se requiriera sin afectar el servicio. La red necesitaba ser capaz de tener conexión a computadoras de diferentes fabricantes y tener la cualidad de que éstas se comunicaran fácilmente con otras.

ARPANET desarrolló un concepto importante sobre el protocolo de red, lo que constituyó un conjunto formal de reglas que las computadoras conectadas a la red utilizaban para comunicarse. Todas las computadoras, no importando el fabricante, tenían que utilizar el nuevo protocolo para tener la capacidad de hablar en una red. Este protocolo de red involucró a una nueva tecnología conocida como *conmutación de paquetes (packet switching)*. Con esta tecnología se cubrieron necesidades importantes como la de hacer más confiable la transmisión de información, enviar más paquetes de información en la misma red, hasta alcanzar su propio destino.

En el año de 1970, los investigadores utilizando la tecnología de conmutación de paquetes comenzaron experimentando con nuevos protocolos de comunicación con la finalidad de proporcionar una red de comunicaciones simple, estable y confiable. Este nuevo conjunto de protocolos de comunicación se conoció como *TCP/IP (Transmission Control Protocol/Internet Protocol)*.

A principios del año de 1980, TCP/IP se convirtió en el protocolo estándar de red para ARPANET. La introducción del nuevo conjunto de protocolos dentro del sistema operativo UNIX *Berkeley Standard Distribution* (BSD) de la Universidad de Berkeley de California, fue la plataforma de "software" para llevar a cabo la interconectividad en universidades (*internetworking*). En esos momentos UNIX BSD era el sistema más utilizado y disponible en las universidades; esto propició que DARPA realizara lo que tenía como proyecto, llevar a cabo la conectividad heterogénea o interconectividad, por lo que muchas organizaciones

que estaban ligadas a ARPANET migraron sus protocolos a TCP/IP y conjuntaron una comunidad mayor.

La red creció en un número impresionante de *hosts* por red en miles y miles de *hosts*. La red original ARPANET se convirtió en la dorsal de una confederación de redes locales y regionales basadas en TCP/IP, conocida como la red Internet, es decir, la nueva red Internet abarcaba a todas las redes con el nuevo protocolo conectadas a ARPANET. En 1983, la conversión de TCP/IP fue completa y el Departamento de Defensa de los Estados Unidos convocó a que todas la redes grandes usarían el protocolo de protocolos: el estándar de TCP/IP.

En 1988 DARPA decidió que el proyecto experimental de red había terminado. El Departamento de Defensa de los Estados Unidos empezó a desmantelar la red ARPANET. En ese año, surge otra red, fundada por la Fundación Nacional de Ciencia (National Science Foundation) llamada NSFNET, integrando a la red ARPANET como la red dorsal de Internet.

En nuestros días, la red Internet conecta a millones de computadoras alrededor del mundo proporcionando una gran cantidad de información que está disponible, convirtiéndose en la más popular. Actualmente permite manipular aplicaciones conocidas como *telnet (conexión remota), ftp (transferencia de archivos), correo electrónico (envio de mensajes a través de la red), gopher (sistemas de información estáticos basados en texto, imagen y audio), wais (indexación de bases de datos para sistemas de información), WWW (sistemas de información de hipermedia interactiva), etc.* La red está basada según su topología, en un modelo abierto donde el protocolo en la que corre está bien diseñado, permite utilizar servicios que facilitan el intercambio de información, el usuario accesa fácilmente y de una forma muy simple.

Lo que en un inicio fue una red diseñada para un grupo pequeño de usuarios de carácter científico, ahora es un potencial enorme en cuanto a la variedad de posibilidades de uso que existen y, es usada por toda clase de personas en el mundo.

El éxito de Internet es el esfuerzo realizado por años de agrupamiento y deseo de establecer un intercambio de información mundial estable; todo esto se ha logrado gracias a la integración de sus servicios que permiten que el usuario manipule la información transparentemente, es decir, sin que tenga que tomar en cuenta los medios de transferencia, para lograr el único fin: compartir información.

1.2 HISTORIA DEL SISTEMA DE NOMBRE DE DOMINIO (DNS)

Inicialmente los nombres de las máquinas en la red estaban siendo administradas para Internet por Standford Research Institute-Network Information Center (SRI-NIC). Se administraba el espacio de nombres y se determinaba si el nombre era apropiado o no para el equipo, nombres prohibidos, obsenos o repetidos eran eliminados o simplemente rechazados.

Durante la década de los setentas, ARPANET era una red pequeña, una comunidad amigable con algunos pocos cientos de *hosts*. La administración de los hosts era por el SRI-NIC y la llevaban en una única tabla (o archivo) llamada HOSTS.TXT, que contenía toda la información de las redes que se conectaban con la red central ARPANET; esta información era necesaria para conocer a los *hosts*. Contenía un mapeo de nombre hacia la dirección para cada *host* conectado a ARPANET.

En UNIX es muy común trabajar con la tabla de *hosts* (en ambienes UNIX en */etc/hosts*), como, proveedor de información de algunas computadoras de la red local o bien, la red que se conecta a otra. La idea fue que el archivo HOSTS.TXT, fuera obtenido por cada red o computadora para ser utilizada en una tabla local como el esquema de los sistemas Unix.

La tabla se mantenía actualizada por el SRI-NIC, mejor conocido como NIC, y se distribuia desde un sólo *host*. Los administradores de ARPANET comúnmente reportaban sus altas, bajas o cambios de nombres vía correo electrónico al NIC, y periódicamente los administradores de redes realizaban una trasferencia de archivos mediante *ftp* y para obtener la tabla actualizada.

Estos cambios en la información eran recopilados en una nueva tabla HOSTS.TXT una o dos veces por semana, la administración se convertía cada día más complicada. El tamaño de la tabla de nombres crecía en proporción al crecimiento de los *hosts*. Sin embargo, el tráfico generado por las constantes actualizaciones de los administradores se incrementaban cada día más. Cada nuevo *host* para adicionar no solo significaba una línea más en la tabla, sino que potencialmente otro *host* se tenía que actualizar en el NIC.

El alto crecimiento en la red, forzó a ARPANET a mudarse a los protocolos de TCP/IP porque comenzaba a convertirse en un estándar de diferentes arquitecturas y además su esquema era muy abierto al crecimiento, con esto la red creció aún más teniendo como consecuencia que la administración de la tabla

HOSTS.TXT se complicara ocasionando problemas tales como: *tráfico y carga de información, colisiones por nombres duplicados y por consistencia.*

Dado que *el tráfico y la carga de información* estaban en exceso, la capacidad de transporte en la red disminuyó y se convirtió en algo incontrolable.

Las colisiones por nombres duplicados fue un problema frecuente debido a que no se tenía con exactitud los requerimientos necesarios para cada red y cuando se duplicaban los nombres o los alias, la red tenía el conflicto de no poder proveer la información correcta. El NIC, proporcionaba las direcciones lo que garantizaba su singularidad en las direcciones y evitaba la repetición en otras máquinas.

La consistencia en el mantenimiento de la tabla a través de la red se convirtió en una labor tediosa y difícil, debido a que si se deseaba agregar un nuevo *host* en la tabla se tenía que esperar la actualización y cuando desde un lugar lejano se quería accesar a ese host nuevo, por razones de cambios dentro de la red local, ese *host* ya había sido dado de baja. Esto ocurría muy frecuente y el trabajo para los administradores del NIC era verdaderamente complicado.

El problema que se observó fue el mal funcionamiento del mecanismo de usar una tabla HOSTS.TXT centralizada. El éxito de ARPANET estuvo basado en que surgió como un experimento y lo que contrastó a este proyecto fue la falla y lo obsoleto de manejar una sola tabla centralizada de *hosts*. Pero, debido a esa inquietud, el cuerpo técnico de ARPANET tomó cartas en el asunto para buscar un sucesor al mecanismo de la tabla única, cuyo objetivo principal fuera el de crear un sistema que resolviera los problemas que la tabla única generaba y que pudiera ser soportado por un sistema de *tabla de hosts* unificado.

Este nuevo sistema debería permitir la administración local de los datos y hacer que la información tuviera un salto global para la disposición de los miembros de la red de forma distribuida.

La descentralización de la red eliminaría el cuello de botella generado por la tabla HOSTS.TXT y el problema de tráfico. Una administración local sería la mejor solución para tener la tarea de estar guardando más fácilmente la información actualizada y mediante algún procedimiento tratar de propagar la información actualizada.

Se usaría un nuevo esquema, el espacio de nombres jerárquico apuntando a nombres de *hosts*. Esto finalmente, aseguraría la unicidad de los nombres.

Finalmente a partir de esas necesidades, Paul Mockapetris del Instituto de Ciencias de la Información de University of South California (USC) fue el responsable del diseño y arquitectura del nuevo sistema. El escribió entonces un nuevo procedimiento *[RFC's 882 y 883 mejorados en RFCs 1034 y 1035]* para presentar una nueva tecnología sobre Internet que permitiera terminar con los problemas de tener una base de datos única, esta nueva tecnología o sistema se le conoce como: *Sistema de Nombre de Dominio (DNS).*

[1]RFC son documentos de Solicitud de Comentarios (*Request for comments*), que son procedimientos informales presentando nuevas tecnologías dentro de Internet. Estos documentos son distribuidos libremente y contienen características y descripciones de la tecnología presentada, generalmente va orientada a desarrolladores.

1.3 FUNCIONAMIENTO DEL DNS

El Sistema de Nombre de Dominio (DNS) es utilizado para traducir nombres de computadoras (*hosts*) a direcciones Internet Protocol (IP) de red, y para traducir las direcciones IP a nombres.

El DNS es una base de datos distribuida que contiene la información de una máquina. Permite el control local de las agrupaciones lógicas (nodos de red) de toda una red. Los datos en cada segmento son disponibles a través de toda la red bajo un esquema de cliente-servidor. La robustez y el adecuado funcionamiento, se alcanzan a través de la optimización y la distribución de datos en la red.

Los diseñadores del DNS al considerarlo como una aplicación de red, en lugar de almacenar los datos centralizados en una sola tabla *HOSTS.TXT*, decidieron distribuirlos en la red, lo que constituyó una gran ventaja que tiene el sistema. Bajo este esquema se pudo obtener que:

- Toda la carga de la red se redujo.
- La carga en un host particular (NIC) se redujo.
- El manejo de la base de datos (administración) se distribuyó.

El DNS emplea el modelo cliente-servidor. El servidor hace todo el trabajo inherente en una cierta tarea; es decir, es el que hace la administración de las bases de información y atiende peticiones. El cliente solicita al servidor una consulta. Los servidores en este caso son programas especializados los cuales se están siempre ejecutando , ya sea en un estado de proceso o en espera de petición o en alerta en las estaciones de trabajo (computadoras), esperando que los clientes los contacten. Los clientes emplean funciones o subrutinas para solicitar al servidor la información necesaria.

En resumen, el servidor es un programa que proporciona información al cliente y está corriendo dentro de un sistema de red para responder a la solicitud en cuanto el cliente lo desee. El cliente es un conjunto de subrutinas que están solicitando al servidor una información determinada.

1.3.1 ESPACIO DE NOMBRES DE DOMINIO

El espacio de nombres, puede ser conceptualizado como una estructura de árbol invertida. Cada rama del árbol corresponde a un dominio o subdominio y cada hoja corresponde a un *host* en particular, generalmente el nivel más bajo del árbol.

En el DNS los datos están indexados por un nombre, es decir, que para reconocer una región o zona dentro de la red es necesario conocerlo por nombre. Es algo análogo al directorio de páginas amarillas telefonónico, existen números de teléfono y asociado a el aparece un nombre de una persona o bien, una compañía.

En el DNS los nombres de dominio tienen una trayectoria que se forma a lo largo de la estructura invertida de árbol (*espacio de nombres*), está trayectoria delimita el nombre de dominio. Este espacio de nombres es similar al sistema operativo UNIX en sus trayectorias absolutas de archivo, es decir, que comienzan a partir de una raíz "/", y en el DNS comienzan a partir de un "." o bien conocido dominio raíz, la diferencia es el orden con que se obtiene una trayectoria, es decir, en el DNS se obtiene del nivel más bajo hacia la raíz y en UNIX de la raíz hacia el nivel más bajo. Ver figura 1-1.

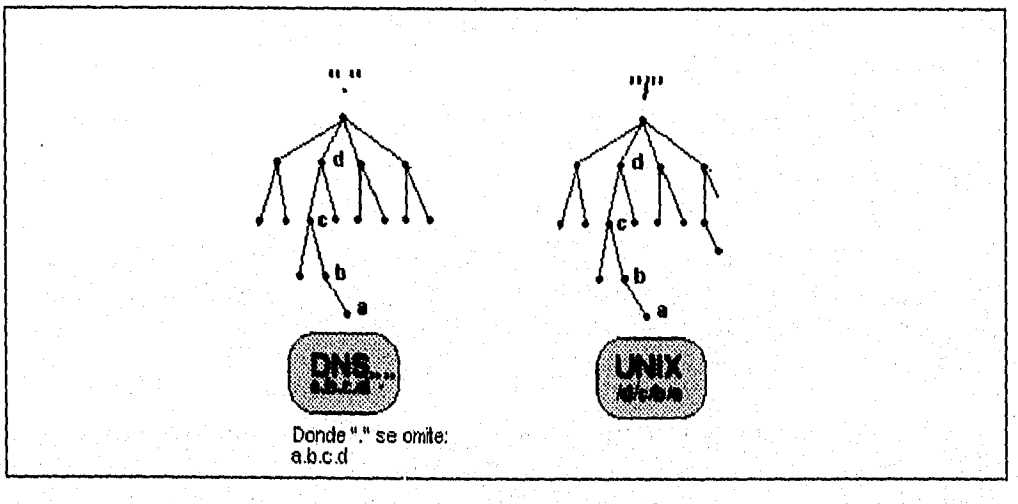


Figura 1-1. Analogía entre DNS y UNIX

NOMBRES DE DOMINIO

Dentro del espacio de nombres de dominio y conceptualizando como un árbol invertido, cada *hoja* o *nodo* equivale a un nombre de dominio, formado por la raíz de dominio que tiene un valor representativo nulo o vacío o simplemente la etiqueta de un punto (".") y cada uno de sus niveles. El nombre completo del nombre de dominio es la concatenación de etiquetas o nombres de dominio separados por puntos desde el nivel más bajo del árbol hasta llegar al tope del árbol (dominio raíz, "."). De aquí es como se obtiene el *FQDN (fully qualified domain name)* conocido como el *nombre de dominio completamente definido.* Ver figura 1-2.

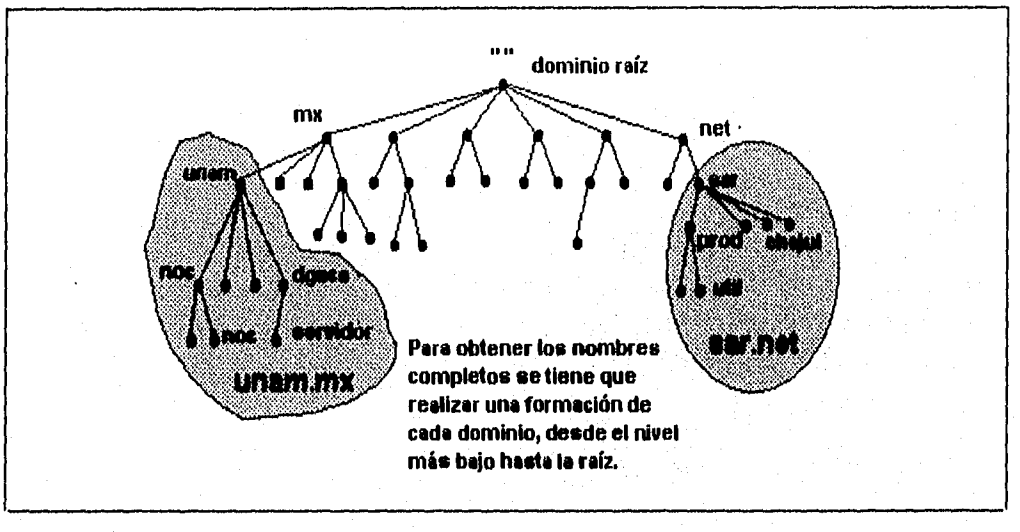


Figura 1-2. Formación de los Nombres de Dominio en el Espacio de Nombres de Dominio

DOMINIO

Un dominio es un subárbol dentro del espacio de nombres de dominio. Un dominio entonces es un nombre o etiqueta que representa a un nodo o concatenación de nodos en el árbol invertido, y hay la posibilidad de dividir el dominio en subdominios.

En un dominio uno puede tener un grupo de *hosts* que representarán ser parte de un dominio y se definirán como *zonas* de dominio. Técnicamente una *zona* y un dominio es lo mismo, pero, existe la diferencia entre ellos. Una *zona* es una región donde pueden estar todos los elementos que se pueden definir en el espacio de nombres y un *dominio* la suma de nodos dentro de la estructura de espacio de nombres.

La diferencia entre el DNS y el NIS, es que en el DNS los dominios son jerárquicos siguiendo la estructura de árbol y la información es distribuida a través de toda la red, en cambio en el NIS la información es centralizada y no existe jerarquía. La información usando NIS se encuentra en una misma tabla que mantiene todos los datos de la red.

El Espacio de Nombres de Dominio en Internet

Los nombres de dominio no tienen reglas para su elección, sin embargo, en Internet existe un espacio de nombres de dominio definido con la finalidad que la administración sea de una forma distribuida. Para tramitar un dominio de nombre es necesario solicitarlo al NIC correspondiente del país en que uno se encuentre. Ver Anexo B de Registro de Solicitudes.

Los niveles o *hojas* en que se ha dividido Internet en Estados Unidos para reconocer a los dominios es el siguiente, ver tabla 1-1.

Dominio	Funcion
com	Organizaciones Comerciales
edu	Organizaciones Académicas y Educativas
gov	Organizaciones Gubernamentales
mil	Organizaciones Militares
net	Organizaciones de Conectividad
org	Organizaciones No Lucrativas
int	Organizaciones Internacionales
arpa	Organización de ARPANET
[2]códigos por país	Para cada país se asigna un código que hace que cada uno haga su estructura deseada, sin embargo, se ha adoptado la notación anterior para reconocer a las organizaciones. Ej. Para México mx, para Canadá ca, para Cuba cu, etc.

Tabla 1-1. Organización de Dominios en Internet

[2]Códigos por país, el NIC utilizó el ISO 3166 que son codigos alfabéticos o códigos numéricos para representar a los nombres de los países. Usando esto el NIC asignó etiquetas o nombres de dominio para cada país en el mundo. Ver en el Anexo B en el Registro de Solicitudes de Dominio la lista de los códigos de países conectados a Internet.

El Dominio IN-ADDR.ARPA o Dominio Inverso

En el DNS existe un dominio especial que sirve para realizar el mapeo inverso o bien de direcciones IP a nombres este dominio es el conocido como ***in-addr.arpa***.

El dominio especial o bien, dominio inverso tiene el mismo mecanismo para obtener un nombre mediante la dirección IP. Tiene una estructura de árbol invertida similar a la estructura de resolución para formar los nombres de dominio. Ver figura 1-3.

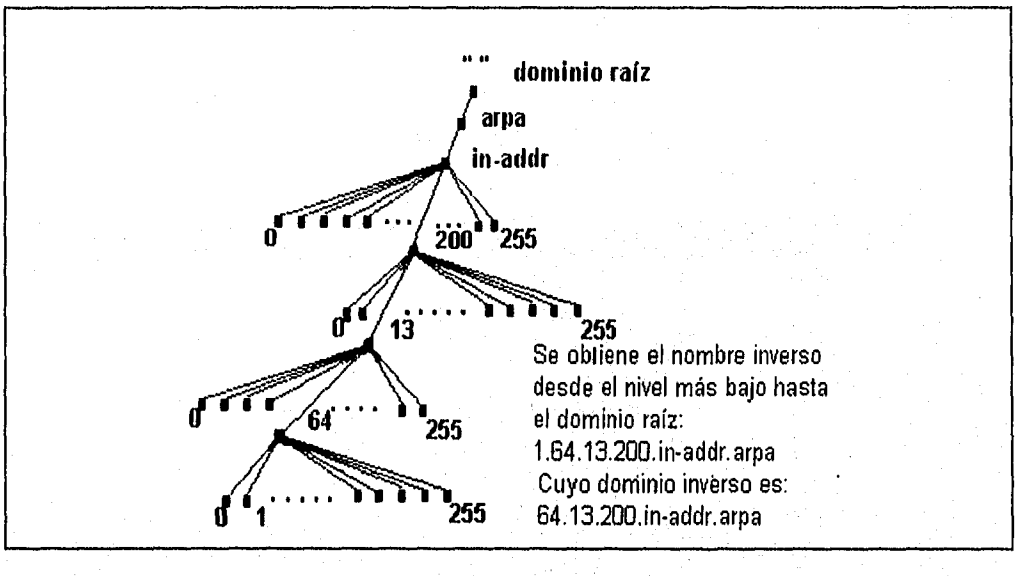


Figura 1-3. Dominio Inverso (*in-addr.arpa*)

El nombre ***in-addr.arpa*** no representa a ninguna organización actual registrada ante el NIC o red limitada, sino que representa a un dominio especial que permite a las direcciones mapear a nombres, como las direcciones son números y no cae en el espacio de nombres, este dominio se forma de manera inversa para poder realizar el mapeo inverso. El dominio de mapeo inverso tiene la característica de que en el dominio *in-addr.arpa* se forma con 4 etiquetas más, éstas corresponden a los 4 octetos de la dirección IP. Se invierte la dirección IP y formará junto con el nombre del dominio especial el nombre completo de *dominio inverso*, el cual está listo para mapearse en los dominios que se forman en el árbol invertido como se pudo apreciar en la figura anterior. Por otro lado, el *dominio inverso* es el que genera el espacio de nombres inversos al determinar una dirección de red (Clase A, B, C) junto al dominio *in-addr.arpa*. Es decir, que cuando se tiene una clase de red determinada, el dominio es el formado por la red

invertida concatenado con el dominio in-addr.arpa. Por ej. se tiene una clase `200.13.64.0` (Clase C), el dominio que se tiene es el de `64.13.200.in-addr.arpa`, el cual tiene la posibilidad de tener 254 nodos dentro del dominio. Cada nombre tiene que estar completamente definido invirtiendo la dirección IP y agregando el dominio especial.

Es importante tener presente la diferencia entre la nomenclatura de los nombres de *hosts* y direcciones IP y recordar que no hay correspondencia entre un campo de nombre de dominio y un campo de dirección IP. Es también posible que tenga varios nombres lógicos apuntados a una sola dirección IP, y a su vez las demás direcciones IP puede pertenecer a otros dominios.

El dominio *in-addr.arpa* tiene un tratamiento especial en el DNS y a su vez de gran utilidad ya que delimita el uso de direcciones autorizadas de cada dominio, y es importante mencionar que para que un *host* sea auténtico dentro de Internet, tiene que declararse en el dominio de nombre y en el dominio inverso o *in-addr.arpa*. En este dominio inverso para realizar la resolución inversa mediante direcciones IP a nombres, los servidores raíz mantienen las bases de datos generales de toda Internet para direcciones validas con información sobre servidores de nombres de dominio que pueden resolver esas direcciones, claramente se puede decir que el dominio *in-addr.arpa* es el espacio de nombres utilizado únicamente para el mapeo de nombres de las direcciones IP a nombres.

1.3.2 DELEGACIÓN DE AUTORIDAD

De acuerdo al espacio de nombres del DNS, el modelo jerárquico está basado en niveles y cada nivel es una autoridad para sus niveles inferiores, para esto un servidor, o resuelve o delega esta responsabilidad a otro servidor de menor jerarquía. Ver figura 1-4.

La delegación de autoridad es lo principal que hace que el DNS trabaje. Cada computadora conoce la dirección de al menos un servidor de nombres. Cada servidor de nombres activo, pregunta a otros servidores para descubrir quién o qué servidor es la autoridad delegada para un dominio en particular. Inicialmente cada servidor conoce por lo menos algún otro servidor: el servidor raíz, o sea , uno que está al nivel jerárquico más elevado del dominio de nombres.

Así el espacio de nombres de Internet es una colección de dominios diferentes, más bien, es una estructura de árbol invertida formada por nombres de dominios. Un dominio es un conjunto o espacio de nombres que representan los *hosts* en una red.

Un sistema que necesita traducir un nombre a una dirección, envía una petición a través de la red a un servidor de nombres designado. El servidor activo hará lo siguiente:

- Resolverá la pregunta inmediatamente, si puede; o
- Mandará la petición al servidor raíz y de ahí bajará de nivel jerárquico mediante un apuntador.

Como alternativa, el cliente puede contactar directamente al servidor más lejano para tratar de obtener la información necesaria . Por ejemplo, si necesitamos encontrar la dirección de `noc.noc.unam.mx`, preguntamos a nuestro servidor local. La pregunta, ¿ cuál es la dirección de `noc.noc.unam.mx`? es aproximadamente como se muestra en la tabla 1-2.

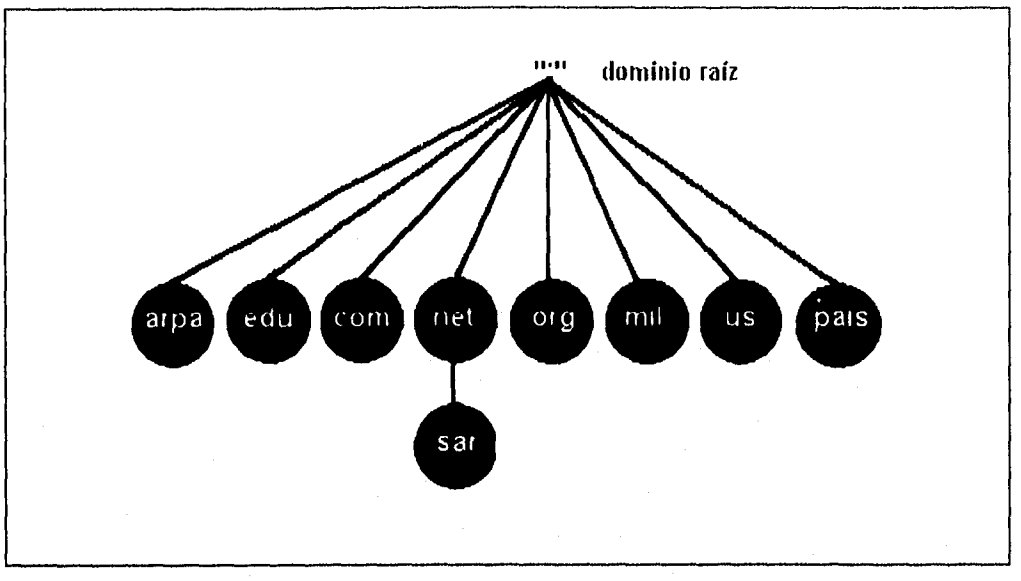


Figura 1-4. Diagrama del Arbol Invertido de Dominios Definidos en Internet

1. ¿ Conozco la dirección de noc.noc.unam.mx ?

2. Si es no, se llevan a cabo los siguientes pasos:

a) ¿ Conozco quién es la autoridad para noc.unam.mx ?
b) Si es no:
i. ¿ Conozco quién es la autoridad para unam.mx ?
ii. Si es no :
A. ¿ Conozco quién es la autoridad para mx ?
B. Si no, contacta el servidor raíz y pide quién es la autoridad para mx .
C. Contacta el servidor para mx y pide quién es la autoridad para unam.mx.
iii. Contacta el servidor para unam.mx y pregunta quién tiene la autoridad para noc.unam.mx
c) Contacta la autoridad para noc.unam.mx y pide la dirección de noc.

3. Regresa la dirección de noc.noc.unam.mx si existe !

Tabla 1-2. Algoritmo de Petición de un Nombre a un Servidor de Nombres

RESOLUCIÓN

En el DNS los servidores de nombres reciben continuamente datos del espacio de nombre de dominio, por lo que deben cumplir con ciertas limitaciones de los clientes. Los servidores no sólo proporcionan información de sus zonas autorizadas, sino que también realizan búsquedas de preguntas a través del espacio de nombres del DNS, a este proceso se le conoce como *resolución.*

Dado que el espacio de nombres está de forma en árbol invertida, lo único que necesita un servidor de nombres es un punto de partida para determinar un nombre no autorizado para él. Necesita apuntar la solicitud requerida a los nombres y direcciones de los *servidores de nombres raíz*. Esto es; que un servidor de nombres puede entonces preguntar a un *servidor raíz* cualquier nombre de dominio dentro del espacio de nombres y si el servidor raíz conoce lo que le están solicitando indica que servidor de nombres tiene la autoridad para responder la pregunta realizada.

Dentro la resolución de nombres, el cliente puede seguir estas formas de resolución:

Servidores Raíz de Nombres. La resolución de los *servidores de nombres raíz* se realiza distribuidamente debido a que ellos conocen a los servidores de nombres de cada dominio en Internet y esto hace posible encontrar la resolución más adecuada a la solicitud. Los servidores raíz mantienen la información de los dominios y servidores de nombres registrados en Internet, que resuelven las peticiones a esos dominios.

Es decir que si reciben una solicitud sobre cualquier dominio, el *servidor raíz* puede dar por lo menos los nombres y direcciones de los servidores autorizados para el nivel más alto del dominio dentro del dominio de nombre preguntado, y aquellos servidores de nombres del nivel más alto pueden dar la lista de los servidores de nombres autorizados por el dominio de segundo nivel del nombre solicitado. Cada servidor de nombres listado proporciona un acercamiento a la búsqueda que se está realizando, pudiendo indicar quien puede responder, o bien dar la respuesta por si misma. Ahora el NIC tiene *servidores raíz* con nuevos nombres, es decir, que temporalmente reajustan administración con el fin de proporcionar un mejor funcionamiento del sistema.

Recursiva. Recursión es solo un nombre para el proceso de resolución usado por un servidor de nombres cuando recibe peticiones de resolución y trata de resolver el dato solicitado recorriendo a través de todo su árbol

jerárquico de autoridadad sobre un dominio determinado. Por este medio encuentra una respuesta favorable o de error. Realiza peticiones recursivas acerca de un dominio en particular, obligando a que el servidor en cuestión obtenga la respuesta o bien mencione que existe la información solicitada. Las preguntas recursivas colocan la mayoría del peso de la resolución en un sólo servidor de nombres.

En este tipo de resolución, un cliente envía una pregunta recursiva para información acerca de un nombre particular de dominio. Al servidor de nombres que se le hizo la pregunta es entonces obligado a responder la información solicitada o bien, indicar un error si el dato solicitado no existe, o que el nombre de dominio especificado no existe. Al servidor de nombres que se le pregunta no puede referirse a otro servidor de nombre por el hecho de se hizo una pregunta recursiva.

Si el servidor de nombres que va a responder no es autorizado para responder la solicitud, tendrá que preguntar a otros servidores de nombres para hallar la respuesta. Este pudiera enviar preguntas recursivas a esos servidores de nombres así los obligará a encontrar las respuesta y a regresarla, o pudiera enviar preguntas iterativas, y posiblemente referirse a otros servidores de nombres más cercanos al nombre de dominio que se está buscando. Comúnmente se realiza una combinación de procesos resolutivos con la finalidad de encontrar la respuesta a la pregunta de un cliente.

Iterativa. La resolución *iterativa* no requiere tanto trabajo por la parte del servidor de nombres utilizado. En esta resolución un servidor de nombres simplemente regresa la mejor respuesta que ya ha resuelto por otro método al cliente, no habiendo una pregunta adicional requerida. El servidor de nombres al que se le está preguntando, consulta sus datos locales o su información local (incluyendo el cache) buscando la infomación solicitada. Es importante mencionar que si no se encuentra la información mediante esta resolución, al menos hará su mejor intento para que encuentre la información que le ayudará al proceso de resolución, usualmente serán los nombres y las direcciones de los servidores más cercanos a la información que se está buscando.

Inversa. Consiste en la información del dominio inverso de *in-addr.arpa*, cuando se pregunta de dirección IP a nombre. Anteriormente explicado.

Cache. Resuelve la solicitud en base a las peticiones anteriores. Esta característica se mencionará más adelante.

En la resolución de nombre se tiene un ***algoritmo general de resolución*** de nombres, el cual se presenta a continuación en las tabla 1-3 y 1-4.

ALGORITMO GENERAL DE RESOLUCIÓN

DEFINICIÓN DE FUNCIONES

Sea que las siguientes variables están definida de la siguiente forma:

nombre; *es el fully qualified domain name (FQDN) que es el nombre completo a resolver (ej. A.B.C.D).*
cuenta(nombre); *regresa el número de campos que tiene el nombre tomando como separador de campo el "."*
fprefijo(nombre), *es la función que devuelve el primer campo de nombre.*
fposfijo(nombre); *es la función que devuelve el último campo del nombre.*
fcorta(nombre, inicio, #elementos); *es la función que corta campos del nombre desde el valor de inicio hasta el número de elementos.*
dominio; *es el dato que se obtiene de los últimos campos de un nombre.*
respuesta; *es el valor de la petición de la resolución.*
fconoce(nombre, búsqueda_servidor); *es la función que pregunta si el servidor de nombres tiene autoridad sobre nombre.*
fresuelve(nombre, búsqueda_servidor); *es la función que obtiene la información que provee el servidor de nombres para el nombre.*
busca_autoridad(búsqueda_servidor, dominio); *es la función que localiza cúal es el servidor de nombres que tiene la autoridad sobre el dominio.*
aleatorio(servidor_raiz); *es una función que devuelve un servidor raiz de forma aleatoria.*

Se inicializan las funciones como:

```
nombre=FQDN=A.B.C.D;
n=cuenta(nombre);
fprefijo(fcorta(nombre, 1, 1);
fposfijo(fcut(nombre, n, 1);
host=fprefijo(nombre);
dominio=fcorta(nombre, 2, n-1);
búsqueda_servidor[]= servidor_local;
```

Tabla 1-3. Algoritmo General de Resolución

Dado lo anterior:

```
1. Se envía una pregunta de un cliente a un servidor local.
m=1
2) Si conoce(nombre, búsqueda_servidor)
        Si resuelve(nombre,búsqueda_servidor)
        {       resultado
                EXIT
        }
        si es no
        {       ERROR
        }
3) Pregunta a un servidor raiz la autoridad para el dominio de la petición.
        búsqueda_servidor[] = { aleatorio(servidor_raiz) }
        ; donde servidor_raiz es un arreglo.

        resultado=respuesta(nombre, n, m,búsqueda_servidor)
        ; n=número de campos y m=1
función respuesta(nombre, n,m, búsqueda_servidor)
{
        Si  n diferente de 0
        {
                Si conoce(nombre, búsqueda_servidor)
                        Si resuelve(nombre, búsqueda_servidor)
                        {       resultado
                                n=0
                        }
                        si es no
                        {       ERROR
                                resultado=0
                                n=0
                        }
                si es no
                        Si busca_autoridad(búsqueda_servidor, fcut(nombre,n,m)
                        {       servidor=búsqueda_servidor }
                        si es no
                        {       ERROR }
        resultado=respuesta(nombre, n-1, m+1, servidor)
        }
        else
                { ERROR; respuesta=0; }
```

Tabla 1-4. Continuación del Algoritmo General de Resolución.

Se puede notar que una pregunta al DNS es *recursiva* por naturaleza: en cada nivel el servidor realiza el mismo tipo de pregunta - dirigido desde abajo, hacia la raíz del árbol en el espacio de nombres - para satisfacer la petición original y eventualmente encontrar un servidor con autoridad para el nombre solicitado.

1.3.3 CACHE

Una de las características que hace más eficiente al DNS y sus servidores de nombres es su capacidad para poder tener un almacenamiento cache de las respuestas a peticiones recientes. Es decir, que la información puede ser reutilizada para responder una petición posterior. La respuesta para cada pregunta incluye otra información que es también guardada en el cache, como también la autoridad delegada para responder preguntas sobre un dominio.

Para asegurar que el cache no se corrompa, una parte de cada registro en el DNS se conoce como "time to live" o "tiempo para vivir" (*ttl*). El campo de *ttl* es de un valor entero y se define en segundos, es el tiempo que puede emplear al cache en un servidor no autorizado antes de ser desechado el cache. Supongamos, por ejemplo, el servidor de nombres `www.xxx.yyy.zzz` tiene un *ttl*=7200, o bien, dos horas. Un servidor se ejecuta sobre el *host* `aaaa.bbb.ccc.dddd` (que no tiene datos autorizados sobre `www.xxx.yyy.zzz`) busca la dirección de `www.xxx.yyy.zzz` y lo almacena en cache. Después de un lapso de dos horas, el servidor de `aaaa.bbb.ccc.ddd` considera el registro corrupto y lo elimina del cache dado que cuando solicitó datos al servidor `www.xxx.yyy.zzz` se le indicó el tiempo de *ttl*. A partir de este momento cualquier nueva petición que se haga al servidor se guardará de nuevo en el área del cache.

La ventaja del modelo cache está implicado por su nombre: si otra solicitud es hecha para la misma información (o similar), no se requiere preguntar remotamente; el servidor local proporciona información que obtuvo en peticiones anteriores. El servidor cache es también un cliente de DNS.

1.3.4 DIRECCIONES ESPECIALES

DNS proporciona pocas direcciones con propiedades especiales. Una de estas es la dirección de *loopback* (dirección de sí misma), la cual es la misma en cada sistema y proporciona un mecanismo estándar para habilitar una máquina para comunicarse con ella misma.

Hay también una clase de direccionamiento que se conoce como *backpointers* (direcciones con apuntadores de dominio inverso); estas direcciones son todas en el dominio ***in-addr.arpa*** y provee el mecanismo por donde cualquier dirección de una máquina dada por número (ej. `W.X.Y.Z.`) puede ser traducida a su correspondiente nombre (ej. `A.B.C.D`).

DIRECCIÓN A SÍ MISMA (LOOPBACK)

La *dirección loopback* es frecuentemente referida por los nombres *localhost* y *loopback*. Este tipo de dirección especial es idéntica para todo *host* o máquina - este siempre es `127.0.0.1` - y se refiere a la máquina propietaria (a sí misma). Es una característica de las máquinas que usan direccionamiento IP (*Internet Protocol*). La dirección *loopback* se define y reserva dentro del esquema de direccionamiento IP, para identificar al equipo mismo y para uso general en diagnóstico interno. Es independiente de su dirección de red real adicional.

La dirección loopback permite aplicaciones para *loop back* (circuito a si mismo) es decir, conexión lógica de red a la máquina a si misma; así, cuando `nombre.dominio` abre una conexión a `localhost.dominio`, abre una conexión con ella misma.

APUNTADORES A DIRECCIÓN (BACKPOINTERS)

Cuando se abre una conexión en la red, la máquina remota conoce solamente la dirección del sistema que inició el diálogo (entabló la comunicación). A veces conociendo el nombre, es suficiente mapearse en el DNS y devolviendo la dirección. El mapeo inverso a través de el dominio *in-addr.arpa* es confiable sobre el mismo mecanismo que usan los servidores de nombres distribuidos, para proveer la búsqueda de información de *dirección-a-nombre*, usando la ***dirección inversa conocida*** como la ***clave*** para buscar en el dominio correspondiente en el dominio *in-addr.arpa*.

Hay diferencias importantes entre dominios de *nombre-a-dirección* y de *dirección-a-nombre* (los nombres o claves formadas en el dominio inverso). Los nombres del DNS empiezan con la información menos significativa (el nombre del *host*) en el campo más de la izquierda, y finaliza con el campo más significativo (el nivel más alto del dominio o tope de dominio). Las direcciones IP, en cambio, empiezan con la información más significativa (el número de la red) y finaliza con el campo menos significativo (la subred y números de máquinas). La clave o pseudo-dominio es construída por la dirección invertida de la máquina dentro del dominio inverso y añadiendo el ***dominio especial*** *(in.addr-arpa)*. Para este tipo de dominio que es diferente al de nombres, tiene la notación especial que toda dirección IP tiene un dominio inverso, que hace única a la dirección con un nombre determinado, de esta forma para denotar e identificar a una dirección IP asociada a un nombre, el nombre en este dominio especial se forma de la dirección IP del *host* en forma inversa y agregando el *dominio in-addr.arpa*. Así, para buscar el nombre de la dirección IP `www.xxx.yyy.zzz`, pregunta al DNS para el nombre `zzz.yyy.xxx.zzz.in-addr.arpa` devolviendo el nombre asociado a esta si es que existe.

1.3.5 TIPOS DE SERVIDORES DE NOMBRES

Existen varios tipos de servidores de nombres que pueden ser configurados de las siguientes formas:

- Como un cliente
- Como un servidor de cache solamente
- Como un servidor maestro (autorizado)

Cada una de estas configuraciones se describirán posteriormente en el Anexo A.

Típicamente, las PCs y sistemas controlados en un dominio de bajo uso se configurarán como clientes (por no tener la capacidad de ser servidor maestro), sin embargo, puede que exista la posibilidad de tener a una PC como servidor dependiendo de la plataforma de software que tenga. Las estaciones de trabajo con mayor capacidad se pueden configurar como servidores de sólo cache y como servidores maestros.

En general, las configuraciones típicas que se conocen para la configuración de servidores de nombres, corren en servidores Unix, debido a que estos equipos traen integrados las aplicaciones de resolución de nombres.

CLIENTE

El cliente se le conoce como *resolvedor* (o *resolver)* que son rutinas y procedimientos que accesan a la información de un servidor de nombres. Los programas que necesitan la información de un espacio de nombres de dominio utilizan un cliente o *resolvedor* que realiza la siguiente secuencia:

- Pregunta a un servidor de nombres
- Interpreta las respuestas, dependiendo del tipo de dato que se pregunta
- Regresa la información a la aplicación que solicita un nombre.

Los clientes o resolvedores son rutinas o librerías que cada sistema trae instalado o bien se compila (para cada sistema) para resolver nombres de aplicaciones como transferencia de archivos *(ftp), conexión remota (telnet), correo electrónico (mail), sistemas de información (como gopher y WWW), foros de interés y noticias (news), pláticas interactivas (irc)* y muchas más.

Para que cada cliente pueda realizar la secuencia antes mencionada, debe conocer al menos los datos siguientes:

- ¿ Cuál es el nombre del dominio local ?
- ¿ Cuál es la dirección del servidor de nombres al que solicitará peticiones?
- ¿ Cuál es el orden de resolución (*hosts,dns o nis*) ?

SERVIDOR DE SÓLO CACHE

Un servidor de *sólo cache (caching-only)* es la más simple configuración de servidor: no hay la necesidad de una administración local. Esto frecuentemente es la más conveniente forma de administrar, permite una configuración eficiente. La diferencia entre el servidor cache y el cliente puede ser explicado por el modelo del cliente:

- Un cliente local pregunta a un servidor de nombres remoto
- El servidor remoto regresa los datos solicitados por el cliente local

Un servidor de cache presenta un nivel adicional de abstracción:

- El cliente local pregunta al servidor local
  - El servidor local pregunta al servidor raíz
  - El servidor remoto regresa los datos al servidor local
- El servidor local regresa los datos solicitados al cliente local
- El servidor local guarda información de forma cache para un uso posterior

Una ventaja de utilizar un servidor de sólo cache, es que los datos que se están utizando se pueden aprovechar debido a la característica cache. Es conveniente mencionar que tener servidor de sólo cache, es cuando no se tengan los recursos ni autoridad para colocar un servidor maestro.

SERVIDOR MAESTRO (AUTORIZADO)

Un servidor maestro es uno que está **autorizado** por un dominio para responder solicitudes de nombres locales y remotos. El servidor maestro para el dominio de *unam.mx*, por ejemplo, es el requerido, habilitado y capacitado para responder peticiones por todos los clientes de la red (incluyendo a otros servidores de otras redes).

El servidor autorizado (*authoritative server*) tiene dos variantes:

- ***Servidores Primarios***, quienes mantienen y administran la información relacionada al dominio. Estos servidores se anuncian como servidores autorizados con la finalidad de que sean los que respondan las solicitudes de nombres homologados dentro de su dominio. Cuando la información sobre un dominio se modifica en cualquier forma de la administración de la red, los cambios son aplicados únicamente en el servidor(es) primario(s).

- ***Servidores Secundarios,*** se derivan de la información de los servidores autorizados típicamente de un servidor primario, es decir, que se obtienen copias de las bases de información de un servidor primario autorizado. La transmisión de recursos de un servidor primario a un secundario se conoce como una *transferencia de zona* (*zone transfer*), se explicará más adelante.

Cada dominio debe tener al menos un servidor primario, y puede tener cero o mas servidores secundarios (para sistemas en Internet, INTERNIC necesita como requisito al menos un servidor secundario). Por motivo a que cada registro en el espacio de nombres tiene un valor de tiempo de vida. Después de este tiempo los datos se consideran viciados e inalcanzables, se recomienda por seguridad de la integridad de los datos que se proveen, que cada lugar tenga por lo menos dos servidores primarios (por consideración de respaldo).

1.3.6 ZONAS DE TRANSFERENCIA

Las zonas de transferencia son el mecanismo por lo que los *servidores secundarios* son proveídos con las copias de las bases de datos de los servidores primarios. Con frecuencia, un servidor secundario contactará a su contraparte primaria y solicitará la recuperación de la información de la zona para la cual provee respaldo.

Del tiempo de arranque o inicio de la transferencia, el servidor secundario realizará peticiones remotas y establecerá contacto periódicamente al servidor primario y solicitará la información de la zona para el dominio determinado. La frecuencia de estas transferencias ocurre dependiendo de la configuración del intervalo de expiración (*expire*) en el registro de *SOA* (*Start of Authority*) para el dominio. Si el tiempo configurado en el servidor secundario ha terminado, es servidor secundario compara su número serial con el número serial del servidor primario y si existe una diferencia (generalmente si es mayor), entoces, una transferencia de zona ocurrirá, generalmente si el número serial del primario es mayor que el del secundario.

Por lo anterior, se puede decir que ocurre esto cuando existen modificaciones en el número serial del servidor primario y el servidor secundario tiene la tarea de verificar este dato teniendo permitido el acceso del servidor primario.

1.3.7 REGISTROS DENTRO DEL DNS

Cada registro consta de una sola línea de texto formada por campos. Cada línea es representativa para el DNS, lo que significa que el administrador debe verificar que cada registro esté bien definido.

CONSIDERACIONES ESPECIALES

Cuando se crean archivos en el DNS es importante considerar algunas situaciones especiales, como son: comentarios, extensión de líneas y uso de tabuladores en lugar de espacios.

- COMENTARIOS

Los comentarios dentro de un archivo del DNS son representados por un punto y coma ";", cualquier texto siguiente al punto y coma será ignorado por el servidor, desde el inicio hasta el caracter de fin de línea.

```
;Esto es un ejemplo de comentario
```

- EXTENSIÓN ENTRE LÍNEAS

La porción de datos de cualquier registro puede ser extendida a través de un separador de línea (así que los caracteres de fin de línea son ignorados) encerrando el campo de datos completo dentro de paréntesis.

Los registros deben ser extendidos a través de varias líneas por el uso de paréntesis; este escritura es el más comúnmente utilizado donde la porción de datos de el registro consiste en una lista, generalmente se usa para un tipo de registro SOA, más adelante se explicará su función.

```
;Ejemplo sobre extensión de líneas sobre los archivos del DNS
@     IN    SOA   noc.noc.unam.mx   root@noc.noc.unam.mx. (
                        92040.501 ; serial
                        43200     ; refresh
                        215600    ; retry
                        604800    ; expire
                        86400 )   ; minimum
```

- USO DE TABULADORES

Para evitar problemas de interpretación de los archivos de configuración al servidor, se aconseja manejar todos las separaciones por tabuladores en vez de utilizar espacios.

DEFINICIÓN DE CAMPOS DE UN REGISTRO

Los registros dentro del DNS tienen un formato general de escritura, la sintaxis general de un registro es:

name	*ttl*	*class*	*type*	*data[data]*

A continuación se presenta una descripción de cada campo de los registros.

- **name**

  En este registro se define el nombre del valor que puede tomar, va desde el nombre del *host* o bien, alguna facilidad como alias o intercambiadores de correo, además de algunos nombres especiales como los que se listan más abajo (. @, *$ORIGIN*, *$INCLUDE*). Se puede que varios registros son permitidos con el mismo nombre. Si el nombre no se especifica y este campo está en blanco, el nombre del registro anterior o previo se asume.
  El campo del nombre puede también ser de uno de las siguientes secuencias especiales:

  . Un punto en el primer campo denota el dominio presente.
  @ El símbolo de arroba o "at-sign" en el primer campo denota la zona presente.

  $ORIGIN

  La clave $ORIGIN es seguida por el nombre de una zona; la zona nombrada se convierte en la zona actual para todas las entidades o hasta el $ORIGIN es modificado.
  Por ejemplo, un solo servidor frecuentemente provee servicio de nombres para múltiples subdominios.

  $INCLUDE

  Esta palabra clave está seguida por un nombre de archivo: el contenido de este archivo es leído inmediatamente dentro de la base de datos, como si ellos hubieran aparecido en lugar de la secuencia de $INCLUDE <filename>, es decir, que esto incluye los datos de un archivo con datos que pudieran estar en la misma base de datos. Lo anterior se hace con el objeto de no cargar mucho a las bases de datos.

- **ttl**

  El *ttl* o (*time to live*) tiempo para vivir, es el número en segundos del registro que un servidor remoto puede mantener o recordar antes de que se considere viciado, y por lo tanto refrescado de el cache del DNS.
  TTL es un campo opcional. Si se deja sin especificar, el default es el valor mínimo (*minimum*) especificado en ese registro de inicio de autoridad (*start of authority*, SOA). Para fines prácticos se utiliza el de default (*minimum*).

- **class**

  La clase (*class*) es generalmente utilizada para conocer el tipo de red. El DNS fue desarrollado para un uso general, y así el campo de clase fue proporcionado también para servidores, utilizado en diferentes tipos de red y protocolos de red, pudiéndose usar la misma base de datos.

- **type**

  Especifica el tipo de registro que manejará. Es importante determinarlo ya que así se conoce los tipos de datos que se manejan dentro del dominio. Más adelante se listarán y explicarán su sintaxis.

- **data**

  Los datos (*data*) para este registro dependen de cada tipo de registro, los datos esperados pueden ser una dirección IP, *host* o un nombre de dominio, una lista o texto libre (para información).

1.3.8 CLASES DE REGISTRO

Dentro de los registros del DNS, conocidos como RR o Registros de Recursos (*RR, Resource Records*) se puede observar que existen clases de registros de acuerdo al tipo de red, a continuación se mencionan las clases de registro que se han utilizado:

CH

Clase CH de Chaosnet, es otro tipo de red que utilizan la estructura de DNS.

HS

Clase HS de Hesiod, desarrollado para el proyecto Athena en MIT (*Massachusets Institute Tecnology*), que es un servicio de información con estructura diferente de DNS.

IN

Clase IN de Internet, es la tipo de red que se está estudiando en este trabajo.

1.3.9 TIPOS DE REGISTRO

Los tipos de registros son los valores que determinan los datos que se manejan dentro de las bases de datos del DNS.El tipo de registro indica los datos que se manejarán en un servidor y lo que el servidor tiene que interpretar.

La configuración correcta de estos tipos de registros determinarán el buen funcionamiento de la resolución a las peticiones de nombres al servidor de nombres.

A continuación se presentan los tipos más utilizados dentro del DNS:

SOA (Start of Authority)

Registro de inicio de autoridad. Este registro designa el inicio o arranque de un dominio o subdominio (*zona*), designa el encabezado inicial de todos los archivos de configuración. Se requieren ciertos campos como el nombre de la zona, la persona responsable de la zona y el numero serial y varios datos que se sirven para el funcionamiento del servidor.

Es muy importante tener en cuenta a este registro dado que tiene en un campo establecido el número de serie o serial, que es un valor que lleva la contabilidad del número de cambios de información al servidor. Para poder realizar el proceso de reconocer que un servidor tiene información nueva, existen la transferencia de zona, que son los servidores autorizados para transferir datos a través de la red. De este registro depende la buena configuración de nuestro servidor.

Los siguientes valores que tiene el tipo *SOA* debe de ser en el orden siguiente, les llamaremos números especiales.

serial

Este es el número serial de la información de la zona y se debe de incrementar cada vez que cualquier información modifica la base de datos. Los servidores secundarios comparan el número serial de sus registros SOA con el número serial en el registro SOA del servidor primario; si éste es más grande, el servidor secundario solicita una transferencia de zona.
Una forma común de usar el número serial es la fecha de la última actualización del archivo(base de datos), representado como YYMMDD (año, mes y día). Un poco más flexible es proporcionar agregando un par de dígitos *nn* o igual a .nn a la fecha. Así, los números seriales 91040103 y 910401.03 indican que la base de datos tuvo su último cambio en Abril 4, 1991 y que el cambio fue la tercera vez del día.

refresh

Este el intervalo de tiempo de expiración (timeout) para los servidores secundarios, en segundos. Un servidor secundario debería contactar al servidor primario en una zona en cada tiempo en segundos (refresh) para comparar los números seriales en los registros de SOA. Si el SOA del servidor primario es 910401.05 y el serial del secundario para la misma zona es de 910401.03, el secundario solicitará una transferencia de zona y recargará la base de datos del servidor primario.

retry

El tiempo que un servidor secundario debería de esperar después de que falló un intento para contactar su primario. Después del tiempo de reintento en segundos ha transcurrido, el servidor secundario deberá intentar de nuevo.

expire

Es el tiempo máximo, en segundos, que un servidor secundario confiará en sus propios datos. Este valor del registro será configurado suficientemente alto para asegurar que un servidor secundario no recupere su base de datos debido a un prolongado tiempo de estar caído un servidor primario.

minimum

Es el tiempo de default del TTL, tiempo para vivir, o información que puede recordar el servidor en un tiempo determinado.

Este registro pasado al formato general queda como:

nombre	*{ttl}*	*clase*	*soa*	*origen*	*correo-administración (datos)*

donde *nombre* es @ el *ttl* es el tiempo para vivir del registro (opcional), la *clase* es el tipo de red, *soa* es el tipo que estamos definiendo, *origen* es el lugar o host que tendrá la información , *correo-administración* es la dirección electrónica del encargado de la administración y *datos* es la información que determinan los números especiales.

A continuación se presenta un ejemplo de como queda representado el tipo de registro *SOA*.

```
;Ejemplo de SOA
@     in    soa   dns.sar.net.      armando@sar.net. (
                        9204.0501 ; serial
                        43200     ; refresh
                        215600    ; retry
                        604800    ; expire
                        86400   ) ; minimum
```

NS (Name Server)

Es el registro de Servidor de Nombres (*Name Server*). Este registro identifica el servidor de nombres responsable para un dominio dado o subdominio. El campo de nombre del registro denota el dominio; los datos (el dato) debe de ser el valor completo del nombre del servidor de nombres para ese dominio.

Pasando este registro al formato general:

{nombre}	*{ttl}*	*clase*	*ns*	*servidor-de-nombres*

donde *{nombre}* es opcional, pero hace referencia a la zona de autoridad o dominio en que se está trabando, *ttl* es opcional, *clase* es in, *ns* es el tipo de registro y servidor-de-nombres es el que tiene autoridad sobre el dominio.

```
;Ejemplo para NS
sar.net            in    ns          dns.sar.net.
; o como {nombre} es opcional escribirlo, como el siguiente renglón:
                   in    ns          solar.sar.net.
```

A (Address)

Es el registro de Dirección. El dato es la dirección IP de la máquina o *host* especificado por nombre.

En el formato general queda como:

nombre	*{ttl}*	*clase*	*a*	*direcciónIP*

donde *nombre* es el nombre del *host*, *ttl* es opcional, clase es in, *a* es el tipo de dato manejado y *direcciónIP* es la dirección que se le está asignando al nombre del host.

```
;Ejemplo para A
chajul          in    a     200.13.64.1
```

CNAME (Canonical Name)

Es el registro de alias o conocido como Nombre Canónico. El nombre es un nombre de *host* pseudónimo, y el dato es el nombre canónico. Los registros CNAME son frecuentemente usados para proporcionar nombres múltiples de máquinas a una sola máquina.

Escrito en su formato general queda:

alias	*{ttl}*	*clase*	*cname*	*nombre-real*

donde *alias* es el pseudónimo o apodo que se quiere asignar, *ttl* es opcional, *clase* es in, *cname* es el tipo de registro y *nombre-real* es un nombre previamente definido en la base de datos.

```
;Ejemplo para Alias
ftp         in    cname       chajul.sar.net.
```

HINFO (Host Information)

El registro *hinfo* es la información del *host* o máquina. El dato es texto de forma libre que describe al *host*; un entrada típica consiste de la información del hardware y del sistema operativo. Por ejemplo para chajul se puede observar de esta forma:

Así quedaría el registro en su forma general:

{nombre}	*{ttl}*	*clase*	*hinfo*	*Hardware*	*S.O.*

donde *{nombre}*es el nombre de un *host* registrado antes, puede ser opcional, *ttl* es opcional, *clase* es in, *hinfo* es el tipo de registro que muestra información,

Hardware es el tipo de arquitectura del *host*, *S.O* es el sistema operativo del *host* registrado.

```
;Ejemplo para información del "host"
noc         in     a      132.248.204.1
; o como esto
            in     hinfo "sun" "unix/sunos4.1"
```

WKS (Well Known Services)

El registro WKS lista los servicios bien conocidos que son proporcionados en una dirección particular, para un protocolo de red en particular.

Como queda en el formato general:

{nombre}	*{ttl}*	*clase*	*wks*	*direcciónIP*	*protocolo*	*servicios*

donde nombre es el nombre del host a definir, puede ser opcional, *ttl* es opcional *clase* es in, *wks* es el registro en estudio, *direcciónIP* es la dirección de Internet de el *host* determinado en nombre, protocolo es el tipo de *protocolo* usado para las aplicaciones y *servicios* es la lista de aplicaciones soportados por el servidor definido en *nombre*.

```
;Ejemplo para WKS
solar in     wks    200.13.64.2 udp   who route tacacsd
      in     wks    200.13.64.1 udp   who route domain
      in     wks    200.13.64.1 tcp   (echo telnet ftp
                                 finger smtp domain)
```

PTR (Pointer)

El registro apuntador es utilizado para mantener los apuntadores de mapeo inverso para el *in-addr.arpa*. El nombre típicamente será la porción del host de la dirección IP; el dato será el nombre completo del host. La entrada para chajul (asumiendo que el $ORIGIN está configurado para `64.13.200.in-addr.arpa`) es asi:

En el formato general queda:

nombre	*{ttl}*	*clase*	*ptr*	*nombre-real*

donde *nombre* es el octeto(s) que definen a la dirección IP del nombre real, por ejemplo para `1.64.13.200.in-addr.arpa`, el dominio para ese nombre es `64.13.200.in-addr.arpa` y en el registro se enumera únicamente el octeto faltante, en este ejemplo será 1, *ttl* es opcional, *clase* es in, *ptr* es el registro apuntador que traduce de dirección a nombre, *nombre-real* es el nombre completamente definido en un registro tipo A.

```
;Ejemplo para ptr
1       in      ptr     chajul.sar.net.
```

MX (Mail eXchanger)

El registro MX será información del *intercambiar de correo (Mail eXchanger)* que se utiliza para proporcionar un "mail drop" correo filtro para los hosts (como son las máquinas PC) los cuales no pueden estar disponibles en la red todo el tiempo. Muchos sitios designan un sistema en particular para servir como un servidor de correo.

En el formato general:

nombre	*{ttl}*	*clase*	*mx*	*preferencia*	*nombre-mx*

donde *nombre* es el nombre de un dominio o nombre de un *host* al cual se puede enviar correo, *ttl* es opcional, *clase* es in, *mx* es el registro que permite realizar esta facilidad sobre el correo, *preferencia* es un valor que tiene cierto peso sobre la elección de varios nombre de mx definidos el de menor valor es el que tiene más peso, *nombre-mx* es el nombre real de un servidor que está definido con tipo A, es importante mencionar que debe estar configurado para recibir y entregar correo.

```
;Ejemplo para mx
cotz            in      mx      10      chajul.sar.net.
sar.net.        in      mx      10      chajul.sar.net.
sar.net.        in      mx      0       solar.sar.net.
```

1.3.10 FORMATO GENERAL DE UN REGISTRO EN EL DNS

Los registros presentados anteriormente son los más utilizados en Internet. Existen otros registros, pero, los anteriores son los más utilizados, además son soportados por la implantación del DNS sobre UNIX conocida como **BIND** (Berkeley Internet Name Domain). Ver Anexo A.

Se presenta un resumen donde se muestran los tipos de registros con sus datos asociados, partiendo de la estructura general de un registro. Ver tabla 1-5.

@	mínimo[3]	IN	SOA	Servidor_primario buzon@correo (serial, refresh, retry, expire, minimum)
dominio/subdominio	mínimo	IN	NS	Servidor secundario (Nombre completo, FQDN)
host[opcional][4]	mínimo	IN	A	DirecciónIP
dominio inverso	mínimo	IN	PTR	Nombre completamente definido (FQDN)
host[opcional]	mínimo	IN	HINFO	OS CPU
host[opcional]	mínimo	IN	CNAME	Nombre real completo (FQDN)
nombre del mx	mínimo	IN	MX	Prioridad_de_intercambio y servidor_de_correo

Tabla1-5. Formato Estándar General de un Registro

Es importante señalar que en este trabajo, se considerarán en la administración de tipos de registros los siguientes : **SOA, NS, A, PTR, CNAME** y **MX** debido a que según la práctica son los tipos de registros más soportados y utilizados en Internet.

[3][opcional], donde el valor puede ser heredable por algún registro anterior.
[4]mínimo (minimum), donde es un valor especial del registro SOA, si se omite, se toma el valor del registro SOA.

2. METODOLOGÍA

2.1 INTRODUCCIÓN

Se ha comprobado que, dentro de las etapas del Ciclo de Vida de los Sistemas de Información, las más críticas y las que garantizan un buen desarrollo e implantación de los sistemas son las etapas de Análisis y Diseño. Es en estas etapas donde nos damos cuenta de cuáles son los verdaderos requerimientos de un usuario y planteamos la mejor forma de entregar un sistema de información confiable y libre de errores.

El objetivo de cumplir con dichas etapas es conseguir que los sistemas cumplan las metas de la Ingeniería de Software: Funcionalidad, Utilizabilidad, Costo-Efectividad y Calidad.

- *Funcionalidad*: Se obtiene mediante la evaluación del sistema, determinando si realiza los requisitos previamente definidos, es decir, que el sistema funcione de manera eficiente sobre máquinas y datos reales.

- *Utilizabilidad*: Se define como el grado de facilidad de manejar y manipular el sistema obteniendo gran desempeño sobre algo funcional y útil. Permite perfilar y simplicar el desarrollo del software, mejora la fiabilidad y reduce el costo del sistema, en sí, el sistema debe diseñarse para que pueda ser reestructurado, modificado y implantado en otros sistemas.

- *Costo-Efectividad*: Debe estar basado en la evaluación de la justificación ecónomica del proyecto. Cada sistema debe brindar el máximo beneficio para evaluar si realmente lo invertido se está viendo traducido en trabajo real.

- *Calidad*: Se puede definir como corresponder con los requisitos funcionales y de rendimiento establecido, así como el desarrollo debidamente documentado y esperando los resultados deseados.

La Ingeniería de Software persigue alcanzar los logros más sustanciales ya que es una disciplina que integra métodos, herramientas y procedimientos orientados para el desarrollo y para el funcionamiento confiable de los Sistemas de Información para el mundo real.

2.2 ADMINISTRACIÓN Y PLANEACIÓN

La administración de proyectos se divide en dos actividades que se interrelacionan:

1. *Planeación de Proyectos*: Planear qué se necesita hacer y organizar los recursos para hacerlo.
2. *Control del Proyecto*: Determinar qué se ha hecho, qué falta por hacer y tomar acciones correctivas cuando surjan problemas.

PLANEACIÓN DE PROYECTOS

La resistencia a planear algo amorfo, cambiable y difícil como un proyecto de desarrollo de sistemas es natural. Muchos mantienen la creencia de que la planeación es equivalente a pérdida de tiempo. Prefieren gastar su valioso tiempo empezando directamente el trabajo. Aquí tenemos dos razones para planear el proyecto:

- Sin un plan no se puede realizar una estimación correcta. Con el plan se sabe quién va a hacer qué cosa en cada período de tiempo, qué es lo que se requiere producir o cuál es el resultado que se desea y qué cosas son imperativas realizar. Estos son los elementos esenciales y básicos para realizar cualquier estimación.

- La planeación puede volverse irritante pero esto no es tan grave ya que detecta obstáculos que pueden afectar a un proyecto en el futuro. Preveer estas dificultades puede evitar gastos innecesarios o tomar caminos erróneos que pueden ser muy dañinos en el futuro.

En el Ciclo de Vida de un sistema la planeación de proyectos es fundamental. La planeación consta de varias actividades:

- DEFINIR LOS PRODUCTOS Y TAREAS DEL PROYECTO

Planear establece tareas para ser administradas en un proyecto. El diagrama de actividades puede transformarse en una ruta crítica de red para la administración de proyectos.

- ASIGNAR A LA GENTE PARA EJECUTAR LOS ROLES

Se determina el reparto y se establece la competitividad de los diferentes individuos asignados a cada uno de los roles:

- Para ver si son suficientemente competentes para alcanzar los niveles requeridos o si se requiere de un entrenamiento.
- Para estimar el tiempo que necesitarán para cumplir ciertas actividades.

El reparto necesita ser revisado a detalle en el plan de trabajo y seleccionado para balancear la carga de trabajo de cada persona.

- ESTABLECER LOS PROCEDIMIENTOS DEL CONTROL DE CAMBIO.

Los procedimientos se establecen para evaluar los cambios requeridos y controlar los cambios permitidos. El medio ambiente y las áreas a los que sirve el sistema pueden cambiar durante el desarrollo del mismo, por lo que los cambios deben ser controlados. Muchos proyectos fracasan porque no se consideró cómo resolver y evaluar los cambios ni cómo incluirlos con el menor impacto posible en el proyecto.

CONTROL DE PROYECTOS

Los procedimientos de control del proyecto detectan problemas que pudiesen provocar pérdida de tiempo antes de que estos ocurran. Las acciones para prevenir tales problemas son:

- Asignar gente que no tiene mucho trabajo para ayudar a los que sí lo tienen.
- Resolver los cuellos de botella.
- Asignar gente de las actividades no críticas a las actividades críticas.

Un proyecto está controlado cuando:

- Cada persona sabe qué es lo que tiene que hacer dentro de las dos siguientes semanas y está de acuerdo en que puede hacerlo en el tiempo permitido.
- Nadie debe esperar que algo ocurra; cada persona está trabajando en la tarea que ayuda al avance del proyecto.

- No existen problemas ocultos - el administrador del proyecto está consciente del progreso de cada uno.
- El administrador de proyectos sabe qué está completo y qué no.

Cada vez que se revisa una actividad se hace una estimación basada en el conocimiento del problema y el trabajo involucrado. Con esto se logra una buena estimación del tiempo restante de un proyecto.

2.3 ANÁLISIS ESTRUCTURADO

El Ciclo de Vida comienza cuando un solicitante, quién es un usuario responsable, solicita el servicio de procesamiento de datos. La respuesta a esta *Solicitud de Servicio* es que un Analista haga un Estudio Inicial el cual es un "vistazo" a los problemas para determinar el beneficio potencial de construir un sistema de información. El producto del Estudio Inicial es un Reporte de Estudio Inicial.

El *Reporte de Estudio Inicial* (el cual menciona un estimado del costo y tiempo requerido para hacer un Estudio Detallado) es revisado por la administración quien tiene suficiente información para asignar una prioridad al proyecto y, si se justifica, autorizar el *Estudio Detallado*. Si puede hacerse algo que dé una solución rápida al problema, tal como usar un lenguaje de cuarta generación para crear un reporte especial, esto queda especificado en el Reporte de Estudio Inicial.

El Estudio Detallado:
- Construye un modelo lógico del sistema actual, ya sea automatizado o manual,
- Menciona los objetivos de negocio que el sistema tendría que cubrir,
- delínea un modelo lógico del sistema que cubriría todos los objetivos y
- produce un estimado en bruto del costo/tiempo para desarrollar y operar el sistema que resuelva los problemas identificados.

El *Reporte de Estudio Detallado* es revisado por la administración, quien toma otras decisiones para seguir adelante, en este caso autorizando el desarrollo de una Definición Preliminar de Requerimientos.

La *Definición Preliminar de Requerimientos* refina los modelos lógicos del nuevo sistema, puntualiza los objetivos y establece las restricciones sobre cualquier diseño físico, al punto donde el Analista pueda ir con un Diseñador de Sistemas técnicamente experto y decirle: "Esto es lo que se requiere que haga el nuevo sistema; deme un Contorno Físico que logre estos objetivos de la manera más efectiva y menos costosa."

La clave es la construcción de un modelo gráfico y lógico del sistema que cumplirá con los requerimientos del usuario. Este modelo lógico, junto con el establecimiento de los objetivos y restricciones del sistema, hace una definición de requerimientos adecuada, la cual tiene las virtudes de:

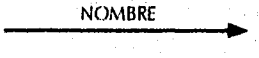
- Expresar qué es lo que se requiere que el sistema haga sin comprometerse en la forma como va a ser implementado físicamente.

- El modelo lógico es una manera muy clara para que el usuario no técnico vea cuál va a ser la naturaleza del sistema y cómo se relacionan sus diferentes partes.

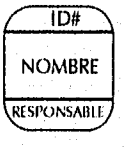
DIAGRAMA DE FLUJO DE DATOS (DFD). Es la representación lógica de un sistema de información que muestra los orígenes y destinos de la información, a través de cuatro símbolos:

- **EE** (*"External Entity"* - Entidad Externa): Representa una fuente o destino de datos fuera del sistema.

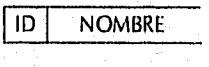
- **DF** (*"Data Flow"* - Flujo de Datos): Es el medio a través del cual se mueven los datos hacia dentro, alrededor y fuera del sistema.



- **PR** (*"Process"* - Proceso): Función del sistema que transforma lógicamente los datos.



- **DS** (*"Data Store"* - Almacenamiento de Datos): Lugar donde se almacenan datos en cualquier forma.



Estos cuatro símbolos son todo lo que se requiere para modelar cualquier Sistema de Información, lo cual puede llevarse al cabo a cualquier nivel de detalle que sea requerido. Partiendo de un primer diagrama general o contextual, cada uno de los procesos contenidos en él se "expande" o "explota" a mayor detalle generando diagramas de menor nivel cada vez más específicos.

Los DFDs son lógicos, es decir, no hacen distinción entre procesos manuales y computacionales, cintas y discos, o cualquier otro tecnicismo. Al ser lógico y usar un conjunto pequeño de símbolos, se entiende fácilmente por usuarios no técnicos. Ver figura 2-1.

A pesar de ser abstractos estos diagramas, son lo suficientemente precisos para ser criticados e ir notando los detalles que hacen falta para, en base a esto, construir diagramas de más bajo nivel.

Estos Diagramas de Flujo de Datos difieren de los diagramas convencionales en dos aspectos cruciales:

- Un DFD lógico no tiene uniones físicas; no muestra tarjetas, cintas, discos, etc. Los mismos símbolos lógicos se usan si el DFD está describiendo un sistema manual o uno automatizado.

- Las flechas en un DFD muestran movimientos de datos; las flechas en un Diagrama de Flujo muestran una transferencia de control. Para saber si estamos viendo un DFD o un Diagrama de Flujo, solo hay que tratar de nombrar los datos que fluyen a través de todas y cada una de las flechas; si no se pueden nombrar, seguramente lo que estamos viendo es un Diagrama de Flujo.

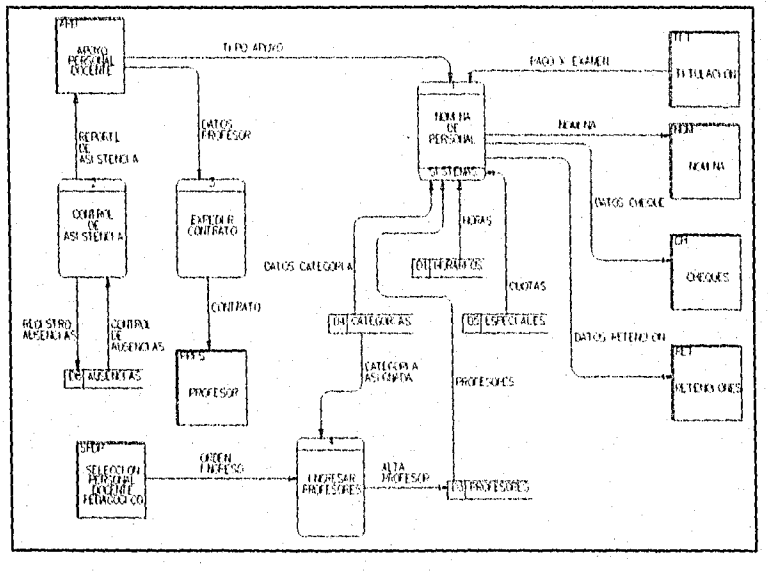


Figura 2-1. Diagrama de Flujo de Datos (DFD)

DICCIONARIO DE DATOS. Al requerirse más detalle de cada flujo de datos, necesitamos ser capaces de expresar su naturaleza lógica.

Las estructuras de datos y elementos de datos se expresan con nombres honestos, completos y significativos seleccionados por el analista. Al llegar al nivel de elemento de datos (un dato que ya no puede subdividirse), se debe especificar su naturaleza lógica (no física).

Si todas las estructuras de datos están compuestas finalmente por elementos de datos, y si podemos: (a) definir cada elemento de datos, (b) establecer la manera en que estos se combinan en estructuras de datos, y (c) establecer qué estructuras de datos se mueven a lo largo de los varios flujos de datos y cuáles se encuentran en los almacenamientos de datos de nuestro DFD. Así, tendremos todos los objetos de datos requeridos para un Diccionario de Datos. Este es el lugar en donde todas las definiciones detalladas de objetos de datos se almacenan.

Ahora hay que considerar la definición de *Almacenamientos de Datos (DS).* Para cada DS hay que definir su contenido (en términos de las estructuras de datos definidas en el Diccionario de Datos). Además, para algunos DS's tendremos que mostrar los accesos inmediatos que tienen que hacerse sobre él.

2.4 MODELADO DE DATOS

OBJETIVOS DE DISEÑO

Se consideran los almacenamientos de datos como la esencia de los sistemas de información (figura 2-2). Siendo los objetivos generales del diseño de la organización del almacenamiento de los datos los siguientes:

- Los datos deben estar disponibles cuando el usuario desee usarlos.
- Los datos deben ser precisos y consistentes (deben poseer una integridad). Más allá de esto, dentro de los objetivos se incluyen almacenamiento, actualización y grabado eficientes de los datos.
- Finalmente, es necesario que el acceso a la información tenga un propósito. La información obtenida de los datos almacenados debe contar con un formato útil que facilite la administración, la planeación, el control y la toma de decisiones.

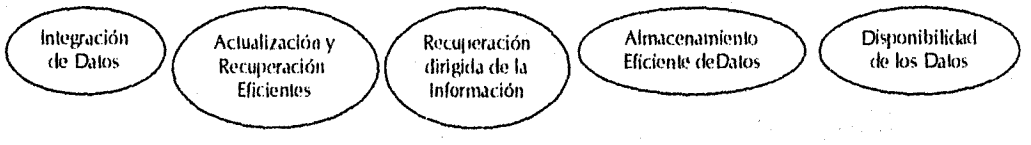


Figura 2-2. Almacenamiento de Datos

ARCHIVOS Y BASES DE DATOS CONVENCIONALES

En un sistema de información se cuenta con dos enfoques para el almacenamiento de los datos. El primer método consiste en almacenar los datos en archivos individuales, exclusivos para una aplicación en particular.

El segundo enfoque para el almacenamiento de datos involucra la elaboración de una Base de Datos (BD). Una BD es un almacenamiento de datos formalmente definido, controlado centralmente para intentar servir a múltiples y diferentes aplicaciones.

ARCHIVOS CONVENCIONALES. Son una manera práctica de almacenar los datos de ciertas (si no todas) las aplicaciones. Se pueden diseñar y elaborar de manera rápida, reduciendo los problemas de disponibilidad de datos y de seguridad. Cuando el diseño de los archivos se realiza de manera cuidadosa, toda la información necesaria queda incluida y se reduce el riesgo de omitir datos de manera accidental.

La velocidad de procesamiento es otra ventaja para el uso de archivos. Hay posibilidad de elegir una técnica óptima para el procesamiento de los archivos de una aplicación sencilla, pero llega a ser imposible alcanzar un diseño óptimo para tareas muy variadas.

El uso de archivos individuales tiene diversas consecuencias. Uno de los principales problemas de los archivos es la falta de potencial para evolucionar.

El rediseño de archivos implica que los programas que los accesan deban redactarse nuevamente de manera acorde, lo cual implica un incremento en el tiempo de programación para el archivo, para el desarrollo y mantenimiento del programa.

Un sistema que utiliza archivos convencionales implicará que los datos almacenados lleguen a ser redundantes.

BASES DE DATOS. Las BDs no son meramente una colección de archivos. Una BD es una fuente central de datos significativos., los cuales son compartidos por numerosos usuarios para diversas aplicaciones. La esencia de una BD es el *Sistema Administrador de la Base de Datos* (DBMS - "Data Base Management System") el cual permite la creación, modificación y actualización de la BD, la recuperación de los datos y la emisión de reportes. A la persona responsable de asegurar que la BD satisfaga los objetivos programados se le denomina *Administrador de la Base de Datos* (DBA - "Data Base Administrator").

Los objetivos de eficacia de la Base de Datos son:

1. Asegurar que los datos puedan ser compartidos por los usuarios, para una variedad de aplicaciones.
2. Que el mantenimiento de los datos sea preciso y consistente.
3. Asegurar que todos los datos requeridos para las aplicaciones presentes y futuras se encuentren siempre disponibles.
4. Permitir que la BD evolucione y se adapte a las necesidades crecientes de los usuarios.
5. Permitir que los usuarios desarrollen su propia visión de los datos, sin preocuparse por la manera en que los datos se encuentran almacenados físicamente.

La lista anterior de objetivos nos advierte las ventajas y desventajas del enfoque de las BDs. Compartir los datos significa que estos deben almacenarse por lo menos una sola vez. Esto a su vez apoya que se mantenga su integridad. Los datos tienen una mayor probabilidad de encontrarse disponibles en una BD más

que en un sistema de archivos convencionales. Una BD con un buen diseño también llega a ser más flexible que dos archivos separados.

Finalmente, el enfoque de la BD tiene la ventaja de permitir que los usuarios expongan sus puntos de vista sobre los datos, sin necesidad de preocuparse de la estructura presente de la BD o de su ubicación física.

La primera desventaja del enfoque de las BDs es que todos los datos se almacenan en un sólo lugar; en consecuencia, los datos son más vulnerables a accidentes y requieren de un respaldo completo. Otras desventajas para la administración de los datos como recurso, se presenta al intentar satisfacer dos objetivos de eficiencia:

1. Reducción del tiempo requerido para insertar, actualizar, eliminar y recuperar los datos en tiempos tolerables.

2. Mantenimiento del costo del almacenamiento de datos en una cantidad razonable.

CONCEPTOS DE DATOS

Antes de considerar el uso de los archivos o el enfoque de las BDs, es importante entender cómo se presentan los datos. Aquellos datos que se obtienen de las personas, de lugares o de eventos de la realidad, eventualmente serán almacenados en archivos o en BDs. Con el fin de comprender la forma y estructura de los datos, se requiere de información acerca de los datos mismos.

ENTIDADES. Una entidad es cualquier objeto o evento acerca del cual se recolectan datos. Puede ser una persona, un lugar o un objeto. También puede ser un evento o unidad de tiempo. Ver figura 2-3.

Figura 2-3. Entidades

RELACIONES. Son asociaciones entre entidades (algunas se refieren como asociaciones de datos, figura 2-4). Existen diferentes tipos de relaciones:

1. El primer tipo de relación es una relación de uno a uno (designada como 1:1).
2. El segundo tipo de relación es una asociación de uno a muchos (1:M).
3. Finalmente, una relación de muchos a muchos (designada como M:N) describe la posibilidad de que las entidades puedan tener numerosas asociaciones en cualquier dirección.

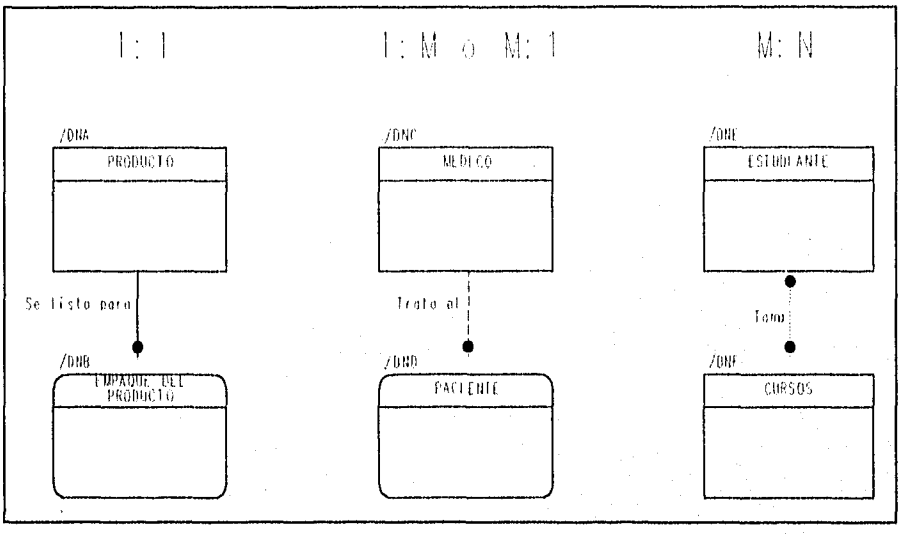


Figura 2-4. Relaciones

ATRIBUTOS. Un atributo es una característica de una entidad. Puede haber muchos atributos para cada entidad. Cuando se elabora el Diccionario de Datos, el elemento más pequeño se denomina "Elemento de Datos" o sencillamente "Dato". Ver figura 2-5.

Los datos son las unidades más pequeñas en un archivo o en una BD. La palabra *dato* también puede utilizarse de manera intercambiable con la de *atributo*. Los datos pueden tener un valor. Estos valores pueden ser de longitud fija o variable; pueden ser alfabéticos, numéricos o alfanuméricos.

En ocasiones, un dato puede referirse como un campo; sin embargo, esto es incorrecto, pues un campo representa algo físico y no lógico. Además, numerosos datos pueden agruparse en un campo; el campo puede leerse y convertirse en numerosos datos.

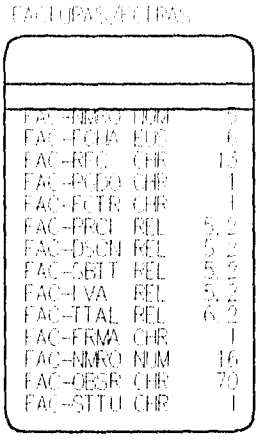


Figura 2-5. Atributos

REGISTROS. Un registro es una colección de datos elementales que tienen algo en común con la entidad descrita. La mayoría de los registros tienen una longitud fija, de tal forma que no es necesario determinar en cada ocasión la longitud del registro. Ver figura 2-6.

Según ciertas circunstancias, se utilizan registros de longitud variable como alternativa para reservar una gran cantidad de espacio para registros más largos.

REGISTRO

NO. ORDEN	Apellido	Inicial	Dirección	Ciudad	Estado	Tarjeta

LLAVE Atributos

Figura 2-6. Registro

LLAVES. Una llave es un dato elemental en un registro, que se utiliza como criterio de identificación para este. Cuando una llave identifica de manera exclusiva a un registro, se le denomina *Llave Primaria* (o criterio primario). La llave primaria identifica la entidad del mundo *real*.

Una llave puede denominarse *Llave Secundaria* (o criterio secundario) si no identifica de manera exclusiva a un registro. Las llaves secundarias se utilizan para seleccionar un grupo de registros que pertenecen a un conjunto.

Cuando no es posible identificar de manera exclusiva un registro utilizando uno de los elementos de datos presentes en él, la llave puede construirse mediante la elección de dos o más elementos de datos combinados. A este criterio se le denomina *Llave Concatenada*.

EJEMPLO DE RELACIÓN ENTRE ENTIDADES

A continuación se presenta un diagrama de relación de entidades. En el ejemplo, un *Médico* tratará a numerosos *Pacientes* (1:M), quienes operan con sus propios *Aseguradores*. Por supuesto, el paciente es sólo uno entre los numerosos pacientes que esperan con el asegurador (M:1). Para completar los registros del Médico, necesita obtener información acerca del *Tratamiento* que ha recibido el paciente. Muchos pacientes reciben numerosos tratamientos (M:N). Ver figura 2-7.

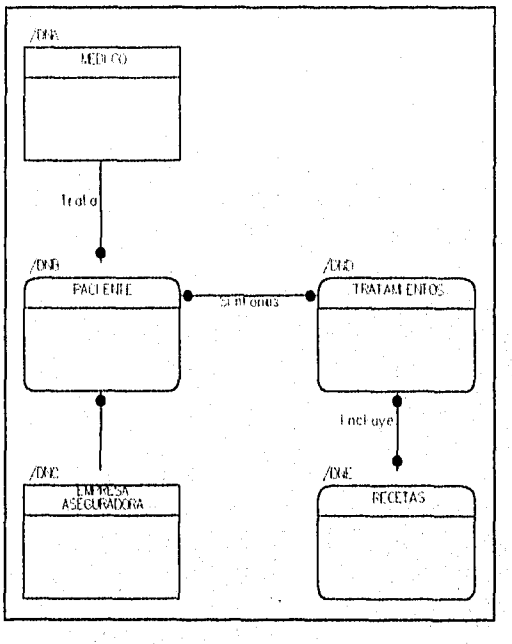


Figura 2-7. Ejemplo de Relación entre Entidades

TIPOS Y ORGANIZACIÓN DE ARCHIVOS

Los archivos pueden usarse para almacenar datos por un período indefinido de tiempo, o bien, pueden usarse como almacenes temporales para un propósito particular.

- *Archivos Maestros*. Contienen registros para un grupo de entidades. Los atributos pueden actualizarse frecuentemente pero los registros en sí se mantienen permanentes. Por ejemplo: registro de pacientes, archivo personal, archivo de refacciones en inventario, etc.

- *Archivos de Tablas.* Contienen datos usados para calcular otros datos o más parámetros de desempeño. Por ejemplo: una tabla de tarifas postales o una tabla de impuestos. Estos archivos generalmente se leen exclusivamente por un programa.

- *Archivos de Transacción.* Se utilizan al introducir cambios para la actualización del archivo maestro. Sólo se introduce la información relevante y necesaria para la actualización. El resto de la información ya existe en el archivo maestro. Una vez actualizado dicho archivo, puede descartarse el archivo de transacción.

- *Archivos de Trabajo.* En ocasiones un programa puede correr eficientemente si usa un archivo de trabajo. Un ejemplo es un archivo que reordena de forma particular los registros con el fin de accesarlos más rápidamente.

- *Archivos para Impresión.* Se usan cuando es necesario registrar la salida de un programa a un dispositivo lógico. El envío de la salida al archivo se llama "spooling". Son útiles, ya que se les puede llevar a otros sistemas de cómputo e imprimirlos en dispositivos especiales.

- *Archivos de Parámetros.* Contienen datos de configuración para determinados programas.Se utilizan cuando se definen características propias de configuración en un sistema. Se leen de forma secuencial con la finalidad de recolectar los datos ahí descritos. Estos archivos no tienen una estructura definida, es decir, cada registro dentro del archivo tiene significado diferente. En este tipo de archivos se pueden actualizar y manipular los datos sobre sí mismos.

En este trabajo de tesis los archivos que se manejan son del último tipo de archivo definido, *Archivos de Parámetros.*

Los archivos pueden estar organizados de diversas formas: *organización secuencial, organización indexada, organización secuencial indexada (índices secuenciales).*

NORMALIZACIÓN

Es el proceso de transformación de las complejas presentaciones de usuarios y de los almacenamientos de datos en conjuntos estables de estructuras de datos de menor tamaño. Además de ser más sencillas dichas estructuras son más estables. Las estructuras de datos normalizadas son más fáciles de mantener.

2.5 DISEÑO ESTRUCTURADO

La fase de diseño consiste esencialmente de dos series de actividades paralelas:

- El diseño de los procesos computacionales (Diagramas Estructurados)
- El diseño de los datos físicos (Archivos / Bases de Datos)

Los resultados de las dos actividades paralelas se unen en la Definición de Diseño, la cual es sujeta a una revisión formal y contiene suficiente información para que el Grupo de Desarrollo se comprometa al horizonte de trabajo dentro del costo/tiempo estimado para programación y pruebas.

Una vez que el modelo lógico ha sido desarrollado por el analista, y la comunidad de usuarios está de acuerdo en que cubre los requerimientos del sistema, se puede comenzar el diseño desde una base firme.

Cuando se ha decidido la frontera de automatización y se haya separado el sistema computacional en Unidades de Diseño (creando el mínimo número de archivos intermedios en el proceso), se tiene que diseñar el software dentro de cada Unidad de Diseño.

Obviamente queremos un diseño que funcione y también que pueda ser cambiado, un ensamblaje cuyos módulos tengan el mínimo de dependencia entre ellos.

Los *Sistemas Modulares* cambiables pueden verse como estructuras de mando militar: el comandante del sistema ordena y recibe información de los subcomandantes, los cuales a su vez ordenan a los módulos trabajadores que hacen realmente las lecturas, escrituras, ediciones, cálculos, etc. Los trabajadores no se hablan entre ellos; nadie habla a menos que se le hable, y habla sólo con su superior o subordinado inmediato.

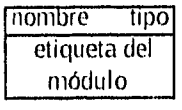
En muchos diseños tradicionales, no estructurados, existen conexiones escondidas o no obvias entre los módulos de manera que si cambiamos una parte, se crea un "bug" en otro lado; al cambiarla para corregir el primer "bug" se crea otro en otro lado y así sucesivamente.

El método más confiable, concebido inicialmente por Constantine, es basarse en el flujo de datos a través del subsistema (aquí está la conexión entre el Diseño Estructurado y el Análisis Estructurado).

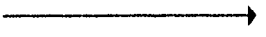
El Diagrama Estructurado muestra qué, cuántos y cuáles módulos controlan a otros módulos haciéndolos funcionar (invocándolos). Ver figura 2-8.

En un Diagrama Estructurado se tienen los siguientes símbolos:

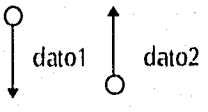
- Los rectácngulos representan los módulos de programación, pueden ser un programa, un subprograma, una función, un texto, etc.;



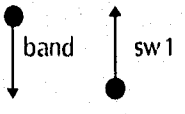
- Las flechas largas representan la "invocación" de un módulo menor por uno mayor;



- Las flechas pequeñas con círculos vacíos representan datos pasados entre módulos, y



- Las flechas pequeñas con círculos rellenos representan información de control (como banderas o "switches").



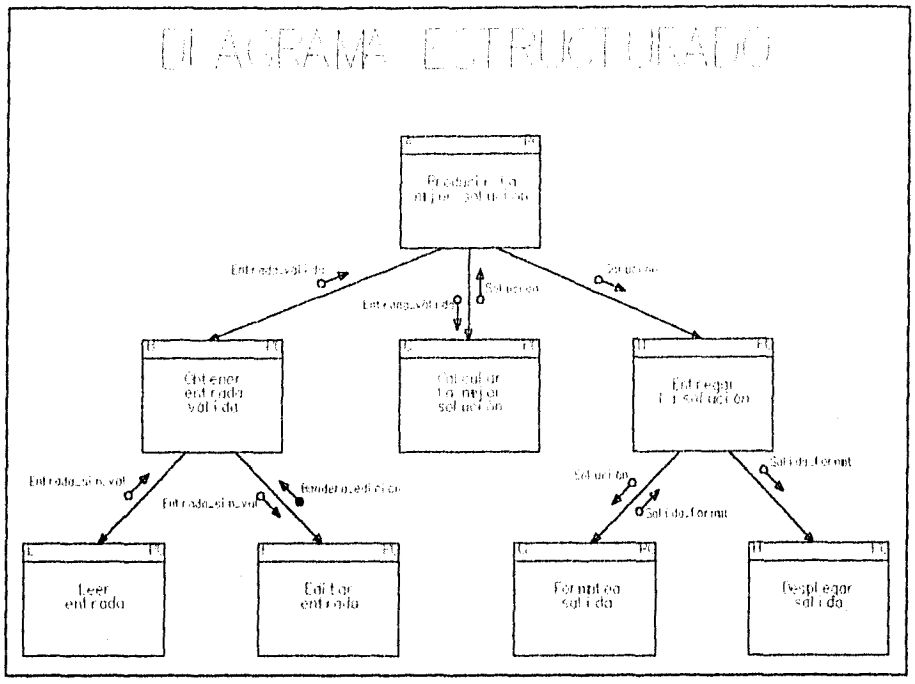


Figura 2-8. Diagrama Estructurado

Los elementos esenciales del Diseño son:

- Diseño de Entradas Eficaces.
- Diseño de Interfaces de Usuario.
- Diseño Efectivo de Salidas.

DISEÑO DE ENTRADAS EFICACES

La calidad de la salida del sistema está determinada por la calidad de su acceso o entrada. Durante el diseño de las formas de entrada y las pantallas es vital tener en mente ésta relación decisiva.

DISEÑO DE FORMATOS. Los formatos son importantes para el desempeño adecuado del trabajo. Son documentos duplicados o preimpresos que requieren ser llenados por las personas en respuesta a un procedimiento estandarizado. Hacen surgir y capturan la información que los miembros de la organización requieren y, con frecuencia, se alimentan a la computadora.

Se deben observar cuatro lineamientos para el diseño de formas, con el fin de que éstas sean útiles:

1. Diseñar formas fáciles de llenar.
2. Asegurarse de que las formas cumplan con el propósito para el cual fueron diseñadas.
3. Diseñar formas que aseguren un llenado preciso.
4. Mantener las formas atractivas.

Diseño de Pantallas. Lo que se mencionó respecto a las formas puede transferirse a las pantallas, pero se deben explotar las diferencias y cualidades únicas del video, y no adoptar ciegamente las convenciones de las formas en papel.

Diseño de Interfaces de Usuario

Una interface es un medio de comunicación entre el usuario y el sistema. La interface permanece como una representación del sistema.

Se debe tener como objetivo el diseño de una interface que ayude a los usuarios y a sus empresas a obtener o introducir información al sistema y que satisfaga las siguientes condiciones (ver figura 2-9):

1. *Eficiencia.* Se demuestra a través de interfaces que mejoran la velocidad de captura de datos y reducen los errores.

2. *Eficacia.* Se logra mediante el diseño de interfaces con que el usuario tiene acceso al sistema de tal forma que sea congruente con sus necesidades particulares.

3. *Productividad.* Se considera el apego a los principios del diseño ergonómico de las interfaces de usuarios y de sus áreas de trabajo.

4. *Consideración del Usuario.* Se demuestra con diseños adecuados de la interface que favorecen la retroalimentación del sistema para los usuarios en forma apropiada.

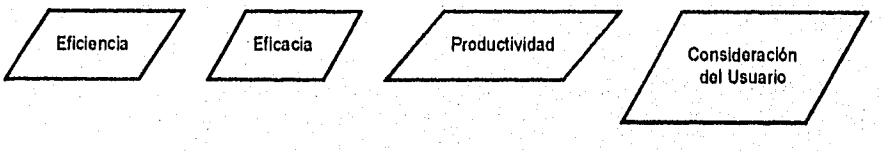


Figura 2-9. Condiciones para el diseño de una interface

DISEÑO EFECTIVO DE SALIDAS

La salida es la información que reciben los usuarios del sistema de información. Antes de convertirse en salida adecuada, ciertos datos requieren de un proceso extensivo; otros sólo se almacenan y cuando se les solicita, se consideran salidas con poco o nada de proceso. Las salidas pueden tomar distintas formas: los reportes impresos tradicionales y salidas en formatos (tales como pantallas en el monitor, microformas, discos, impresora, archivos y salidas de audio).

2.5.1 PROTOTIPAJE

Ahora que se cuenta con Lenguajes de Cuarta Generación (4GL) se ha hecho posible desarrollar modelos de demostración de sistemas muy rápidamente, en días o semanas. Los *prototipos* de demostración son versiones del sistema requerido, con funciones externas, diálogos y reportes aparentemente idénticos al sistema deseado (al menos en parte) con las siguientes características:

- Manejo de volumen limitado para cumplir con los objetivos del sistema.
- Omisión de procesamiento de algunas transacciones o tipos de excepciones.
- Bases de Datos incompletas sólo para demostración y validación.

El objetivo del prototipaje es ejemplificar las funciones del sistema a la comunidad de usuarios, dándoles una experiencia realista de cómo será trabajar con el sistema. Así, la Definición de Requerimientos podrá ser terminada en base a experiencias concretas.

2.5.2 PRUEBAS

La evaluación se debe llevar a todo lo largo del desarrollo del sistema (no sólo al final); cumple con el propósito de identificar aquellos problemas desconocidos.

Conforma una serie de pasos que ayudan a garantizar la calidad del sistema eventual. La evaluación se lleva a cabo conforme progresa el trabajo en los subsistemas o módulos del programa, y se realiza a diferentes niveles y a varios intervalos, aún antes de que el sistema entre en operación.

Todos los programas deben examinarse en cuanto a su diseño con datos de prueba y verificar si los módulos se enlazan entre sí, tal y como fue planeado. También debe probarse el sistema trabajando como una unidad.

Esto incluye la evaluación para las interfaces entre los subsistemas, la operación adecuada de la salida, la utilidad y comprensión de la documentación del sistema y de la salida.

Los programadores, analistas, operadores y usuarios juegan diversos papeles en los diferentes aspectos de la evaluación. Ver figura 2-10.

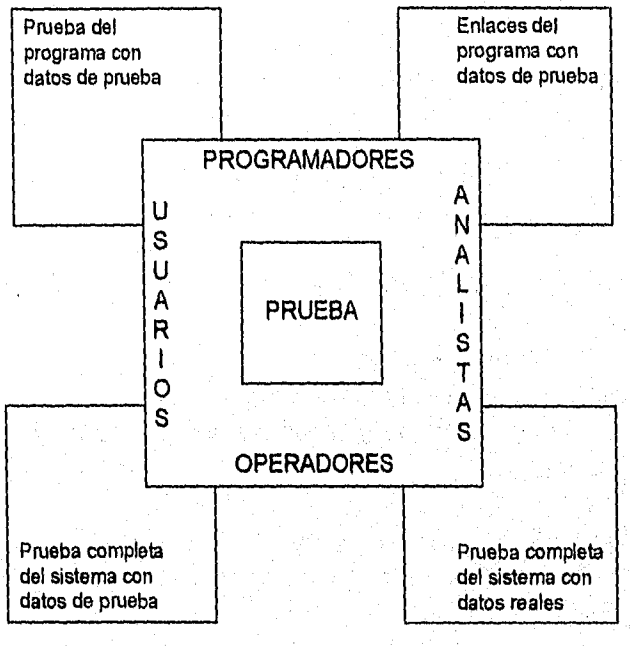


Figura 2-10. Evaluación de Sistemas

2.5.3 DOCUMENTACIÓN

La documentación describe la manera de utilizar un programa, la razón por la que se escribió, las técnicas usadas en su construcción, y aclara cualquier aspecto oscuro relacionado con él. Es un tema que suele ignorarse en los textos de programación.

La documentación puede clasificarse como:

- *Documentación del Usuario.* Se compone de aquellos documentos relacionados con las funciones del sistema, sin referirse a la forma de aplicarlas.

- *Documentación del Sistema.* Describe todos los aspectos del diseño, implantación y pruebas del sistema.

La información proporcionada junto con el sistema debe satisfacer varios requisitos. La documentación tiene que describir:

1. Cómo usar el sistema. Sin esta documentación, aún el sistema más simple resulta inútil.
2. Cómo instalar y operar el sistema.
3. Los requisitos y diseño de todo el sistema.
4. La aplicación del sistema y los procedimientos de prueba para poder darle mantenimiento.

La documentación proporcionada con un sistema puede ser útil en cualquier etapa del Ciclo de Vida de éste. No necesariamente debe producirse en el mismo orden que el sistema mismo.

DOCUMENTACIÓN DEL USUARIO

Suele ser el primer contacto de los usuarios con el sistema, por lo cual debe proporcionar una visión inicial precisa del mismo:

- No debe destacar en exceso las características novedosas o poderosas del sistema, ni debe ser poco realista acerca de sus capacidades.
- No debe ser indispensable que el usuario lea la mayor parte para encontrar cómo utilizar de forma sencilla el sistema.
- Debe estructurarse de forma que el usuario pueda leerla con el grado de detalle apropiado a sus necesidades.

Hay al menos *cinco documentos* que pueden considerarse bajo el encabezado de Documentación de Usuario:

1. Una **descripción funcional** sobre lo que puede hacer el sistema.
2. Un **documento técnico** que explique cómo instalar el sistema y adecuarlo para configuraciones particulares de hardware.
3. Un **manual introductorio** que explique, en términos sencillos, cómo iniciarse en el sistema.
4. Un **manual de referencia** que describa con detalle las ventajas del sistema disponibles para el usuario y cómo se pueden usar.
5. Una **guía del operador** que explique cómo debe reaccionar ante situaciones inesperadas surgidas mientras el sistema se encuentra en uso.

DOCUMENTACIÓN DEL SISTEMA

Se refiere a todos los documentos que pertenecen a la aplicación del sistema, desde la especificación de requerimientos hasta el plan de pruebas de aceptación final. Los documentos que describen el diseño, aplicación y pruebas de un sistema son esenciales si se quiere comprender y dar mantenimiento al programa.

Es importante estructurar la documentación con visiones generales que guíen al usuario hacia descripciones más formales y detalladas de cada aspecto. Los documentos que la componen deben incluir:

1. La definición de requerimientos y, quizás, una fundamentación asociada.

2. Una especificación general de los sistemas que muestre cómo se descomponen los requisitos en un conjunto de programas interactuantes (éste documento no se requiere cuando el sistema se aplica por medio de un sólo programa).

3. Por cada programa del sistema, una descripción de cómo se descompone tal programa en componentes y una declaración sobre la especificación de cada componente.

4. Por cada unidad, una descripción de su operación, que no necesita extenderse a la descripción de acciones del programa, pues tales acciones se deben documentar usando comentarios dentro del programa.

5. Un plan de pruebas amplio que describa cómo se prueba cada unidad de programa.

6. Un plan de prueba que muestre cómo se efectuó la prueba de integración; esto es, la prueba de todas las unidades y programas juntos.
7. Un plan de pruebas de aceptación, diseñado junto con el usuario del sistema. Este plan debe describir las pruebas que es necesario pasar antes de aceptar el sistema.

2.6 EXPECTATIVAS

Finalmente, se puede mencionar que la metodología usada en este trabajo corresponde directamente a las definiciones de conceptos aquí expuestos. El Análisis y Diseño del sistema son las partes más importantes del Ciclo de Vida de los sistemas y es por ello que en este capítulo se describen las características de estas fases de forma detallada con la finalidad de realizar cada una de ellas para obtener la concretización del sistema.

Respecto a la revisión de cada etapa, se pretende que la realización del sistema se haga conforme a lo planeado, analizado, diseñado y modelado. Se haga realmente uso de las herramientas planteadas aquí y sobre todo, obtener el máximo rendimiento del producto obtenido.

3. Análisis y Diseño del SADNS

3.1 Contexto

La automatización del Sistema de Nombre de Dominio (DNS) fue una idea que surgió en el departamento de Redes y Telecomunicaciones (actualmente Subdirección de Redes) de la Dirección General de Servicios de Cómputo Académico (DGSCA) de la UNAM, para proporcionar una mejor administración del DNS.

Inicialmente, la administración del DNS en REDUNAM era una tarea sencilla, sin embargo, a medida que REDUNAM crecía, llegaba a todas sus dependencias y más aún cuando los sectores privados decidieron empezar a conectarse a REDUNAM para obtener las ventajas de Internet, se necesitaba que cada institución diferente a las de la UNAM tuvieran un servidor de nombres funcionando correctamente.

Debido a que los nuevos usuarios conocían poco de los servicios básicos de configuración, REDUNAM prestaba espacio y configuraba los dominios de sus nodos en su servidor de nombres. Poco a poco, los clientes externos de la REDUNAM iban aprendiendo a utilizar y a configurar sus servicios de Internet, por lo que la administración del DNS en la UNAM llegó a complicarse cuando las dependencias comenzaron a dar de alta, baja o cambio a sus equipos dentro del DNS.

Dada esta problematica surgió el presente proyecto, cuyos fines son el mantener las bases de datos de los dominios que tiene la UNAM, así como proporcionar un procedimiento que contenga las facilidades que solo un usuario (superusuario) puede realizar en una máquina. Debido a que el DNS es un sistema que funciona sobre Internet y cuya característica principal es que el servicio que provee es distribuido, ¿por qué no pensar en un sistema que realice administración distribuida a los dominios que están en un lugar físico?.

El problema de la administración de los servidores de nombres no solo se da en la UNAM, sino que es más general. A medida que la explosión tecnológica y comercial de Internet sigue avanzando, los nodos de Internet requieren de herramientas capaces de soportar sus configuraciones básicas y es así como surge el *Sistema de Automatización del DNS (SADNS)* que sirve para administrar el servicio del DNS para cualquier nodo de Internet, proporcionando al administrador de una

red, la capacidad de realizar el mantenimiento de las bases de datos de sus dominios, sin ser un experto en UNIX y en redes TCP/IP.

Actualmente el DNS de la UNAM es administrado por la Subdirección de Redes y Telecomunicaciones de la DGSCA, cuyo conjunto de bases de datos y configuración del servidor de nombres está sobre una estación de trabajo SUN con sistema operativo Solaris 2.4. Para dar mantenimiento a la información relacionada con los subdominios (de las dependencias), se tiene que hacer contacto con la Subdirección de Redes, solicitando el servicio a través de:

- Correo electrónico
- Teléfono
- Fax

El administrador de una dependencia tiene que solicitar a la Subdirección los movimientos que desea y confirmar que realmente se han hecho. Existe por lo tanto centralización en el manejo de la información y además el tiempo de respuesta es lento, dado que todo se hace manualmente de la siguiente manera:

- Editar base de datos correspondiente al administrador de dependencia.
- Cambiar el número inicialización de la base de datos.
- Comparar que el dato solicitado esté dentro de la base de datos respectivas.
- Guardar los cambios en la base de datos respectiva.
- Determinar si la operación fue correcta, si no, se notifica al administrador solicitante.
- Restaurar el servidor de nombres.
- Probar los cambios realizados.

Realmente las tareas son rutinarias y por ello se justifica la realización de un sistema que pueda automatizar esta información, tanto para una administración local, como para una administración remota de cada dependencia accesando a través de la red.

3.2 REQUERIMIENTOS

La idea original fue la de administrar el DNS de REDUNAM, pero, dado que las necesidades de los administradores de nodos de Internet crecían, también se pensó en que otros administradores lo pudieran manejar y utilizar. Este sistema de automatización pretende ser un procedimiento para la administración del DNS, tanto para responsables de redes de REDUNAM como para diferentes dominios de redes existentes que usen el DNS, de manera que se plantea una solución de software que permita manejar las capacidades del DNS.

3.2.1 ADMINISTRACIÓN DEL DNS

La administración del DNS consiste en realizar las siguientes tareas:

1. Inicialización y recarga de datos del servidor.
2. Agregar, borrar, modificar y obtener reportes sobre dominios.
3. Agregar, borrar, modificar y obtener reportes sobre hosts.
4. Solicitar información sobre responsables de red.
5. Agregar, borrar, modificar y obtener reportes de los responsables
6. Llevar una bitácora.
7. Obtención de respaldos sobre hosts, dominios y usuarios.

Debido a que la administración del DNS realiza diferentes tareas manuales, se decidió trabajar sobre un producto de software que tuviera la capacidad de automatizar la administración del DNS.

El proyecto SADNS entonces cobra vida para realizar automáticamente las tareas que un administrador experimentado puede hacer. Uno de los problemas a los que se enfrenta uno diariamente es que los sistemas están basados en interfaces muy complicadas que es necesario tener a una persona técnica para poder manejarlos, pero, SADNS está diseñado para que cualquier persona con el simple manejo de la interface pueda realizar las tareas de administración.

El producto SADNS está orientado para realizar las siguientes necesidades:

- Administración del DNS
- Modos de Administración
- Probar dominios administrados
- Seguridad
- Portabilidad sobre TCP/IP

- Acceso a través de diferentes tipos de interfaces usando el mismo protocolo
- Funcionamiento las 24 horas del día.
- Multiacceso
- Estadísticas de acceso

Administración del DNS

El sistema será capaz de administrar el DNS, es decir, realizar las tareas que el administrador realizaba manualmente, así como proporcionar una interfase amigable que permita que el sistema sea fácil para cualquier usuario, sin necesidad de que éste sea un experto.

Modos de administración

Debido a que la administración la podrán realizar administradores responsables de la información de sus dominios, junto con un administrador maestro, responsable del servidor de nombres, se debe separar la administración en dos modos de administración: modo *administrador-sadns* (maestro) y *administrador-red* (responsable de uno o más dominios).
Quiere decir que existirán dos modos o niveles de autoridad sobre el sistema, que permitirá administrar el sistema de un forma muy eficiente.

Probar los dominios administrados

El SADNS tiene la característica de probar los cambios realizados cuando ha accesado al sistema, mediante una herramienta que está dentro del sistema.

Seguridad

Debido a que el sistema tiene los modos de administración (niveles de acceso), permite que el administrador se autentifique para modificar sus dominios administrados. Se propone un firewall que tenga la capacidad de hacer mejor a este sistema.

Portabilidad sobre TCP/IP

El sistema está sobre el conjunto de protocolos de TCP/IP. Desde cualquier lugar de Internet uno puede accesar soportando este protocolo.

Acceso usando diferentes tipos de interfaces utilizando un mismo protocolo

Debido a que el sistema está corriendo sobre una interface que soporta WWW, existe una enorme lista de clientes que permiten accesar a este sistema, únicamente conociendo la dirección de donde se conecta y las claves y contraseñas correspondientes.

Funcionamiento las 24 horas del día

El servicio de mantenimiento del SADNS será de tiempo completo. Estará siempre a disposición de los administradores. El sistema podrá tener alguna interrupción si se está dando mantenimiento al SADNS o a la máquina que los soporta.

Multiacceso

Al sistema pueden accesar varios administradores simultáneamente con la finalidad de realizar sus modificaciones a sus bases de datos (si tienen administración de algún dominio).

Estadísticas de accesos

El sistema proporcionará estadísticas de accesos de los administradores, obteniendo datos como: ¿quién accesó ?, ¿cuándo ? , ¿a qué acceso?

Utilizando este sistema, la administración distribuida tiene un papel importante, debido a que la administración del DNS es compartida y además permite que las fuentes de información lleguen al sistema de diferentes formas:

- Por cada administrador, conectándose a través de la red
- Administrar su propia información cuando ellos la requieran
- Solicitar algunas cosas especiales al *administrador-SADNS* (maestro) a través de correo electrónico
- En caso de no tener red, tiene que solicitar al *administrador-SADNS* que le administren sus dominios

3.2.2 PROYECCIÓN DEL SADNS

Con la utilización de este sistema, se pretenden obtener estas características y ventajas que son metas perseguidas por la creación del sistema:

- Administración compartida y distribuida (en toda la red)
- Obtención de reportes
- Respaldos de dominios
- Consultar la información los equipos autorizados
- Tiempo de respuesta
- Ampliar conocimientos sobre el DNS

El SADNS es un sistema que pretende incorporar las herramientas existentes en Internet para realizar sistemas interactivos a través de la red, es decir, que se pueden accesar los sistemas a través de la red y mediante sistemas interactivos (de acceso en red) uno puede realizar tareas tales como: registros a eventos, compras interactivas, paseos virtuales, administración de sistemas UNIX (alta, baja, cambio y modificación a usuarios) y por supuesto administración del DNS.

3.3 DIAGRAMA DE FLUJO DE DATOS (DFDS)

Los Diagramas de Flujo de Datos (DFDs) son las herramientas gráficas que nos sirven para realizar el análisis del sistema. Propiamente dicho, la forma de expresar gráficamente el flujo la información en cada módulo a través del sistema es lo que proporcionará la información detallallada para proponer y realizar el programa.

A continuación se presentan los DFDS del SADNS divididos por niveles de abstracción. Los DFDs que se muestran son cinco, que se dividen en el Diagrama General del SADNS (Nivel 0) que contiene a dos procesos (1 y 2). El proceso 1 se expande a varios niveles de abstracción en niveles 2.1, 2.2 y 2.3, donde en cada nivel se generan más procesos que traen como consecuencia entrar a niveles más detallados de cada módulo.

Es importante mencionar que diagramas de flujo de datos son diferentes a los diagramas de flujo convecionales, debido a que los DFDs describen flujos de información de entrada, procesamiento y salida, en cambio en los flujo de datos se describen procesos lógicos de información, se detallan los procedimientos a seguir ante cualquier situación.

En cada DFD no se refleja los necesario para describir un requisito del módulo a programar, debido a esto es necesario tener el diccionario de datos que más adelante se muestra para tener la colección de descripciones de los elementos que forman los DFDs.

Se presentan los DFDs siguiendo con la metodología de Gane y Sarson. Los DFDs se dibujaron con PROKIT (herramienta de metodología CASE), con la finalidad de obtener mayor eficacia en el análisis del sistema.

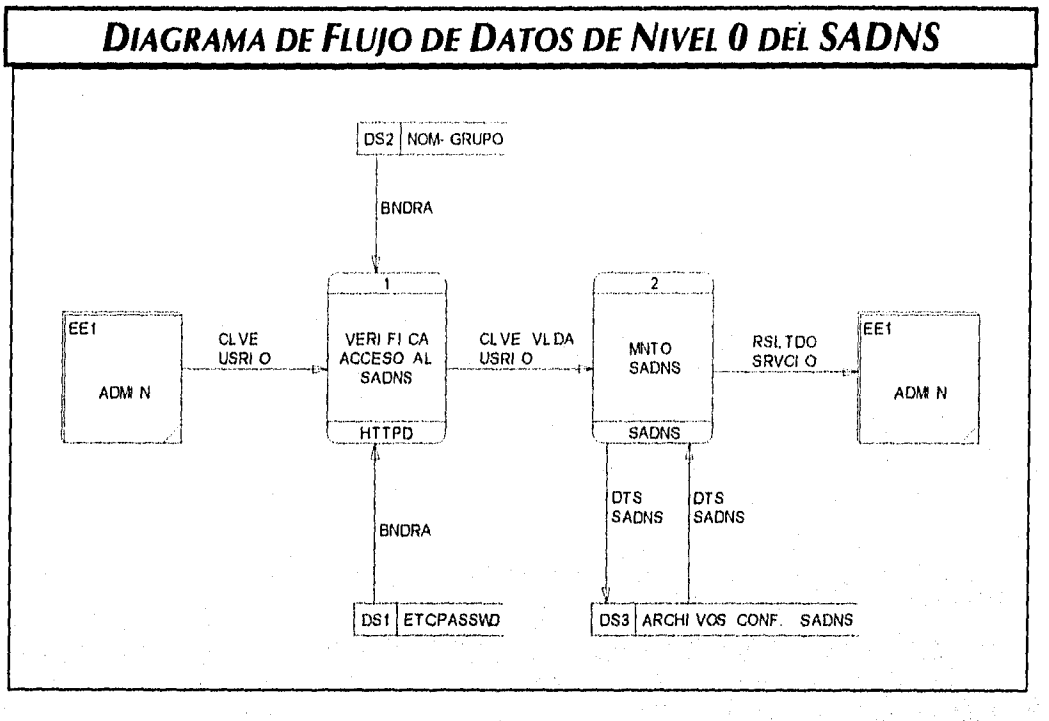


Figura 3-1. DFD Principal del SADNS

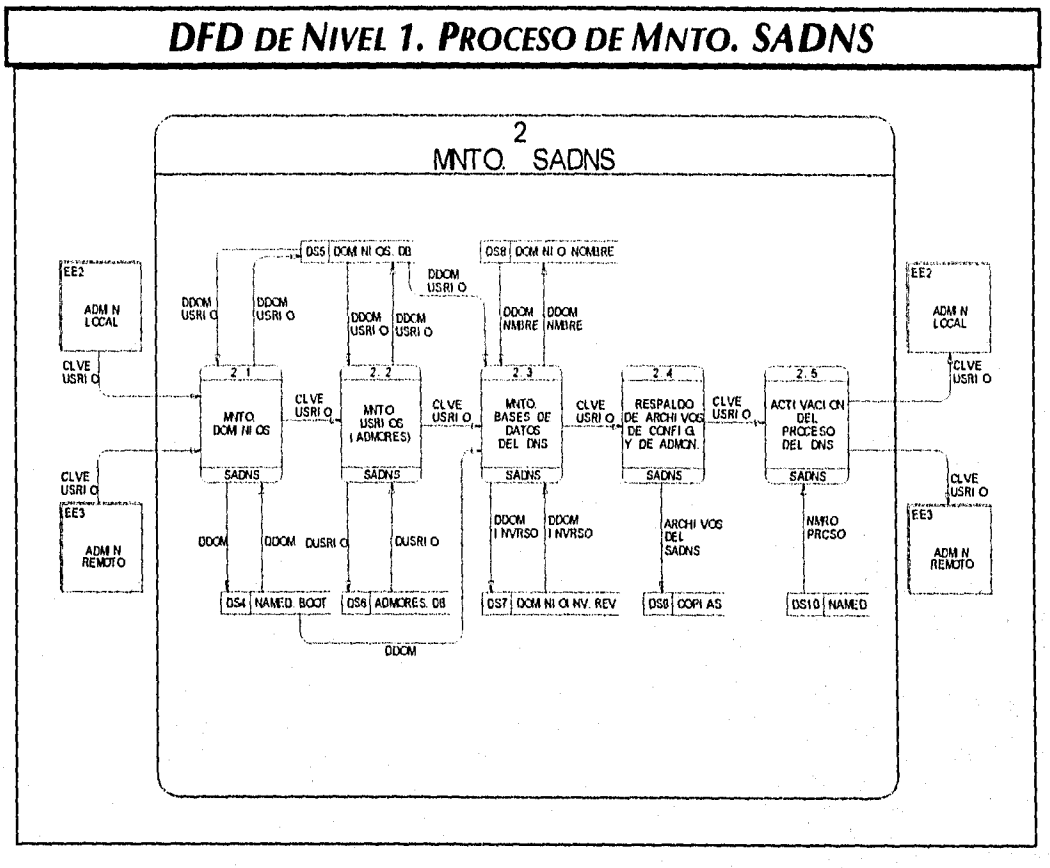


Figura 3-2. DFD del Proceso de Mnto. del SADNS

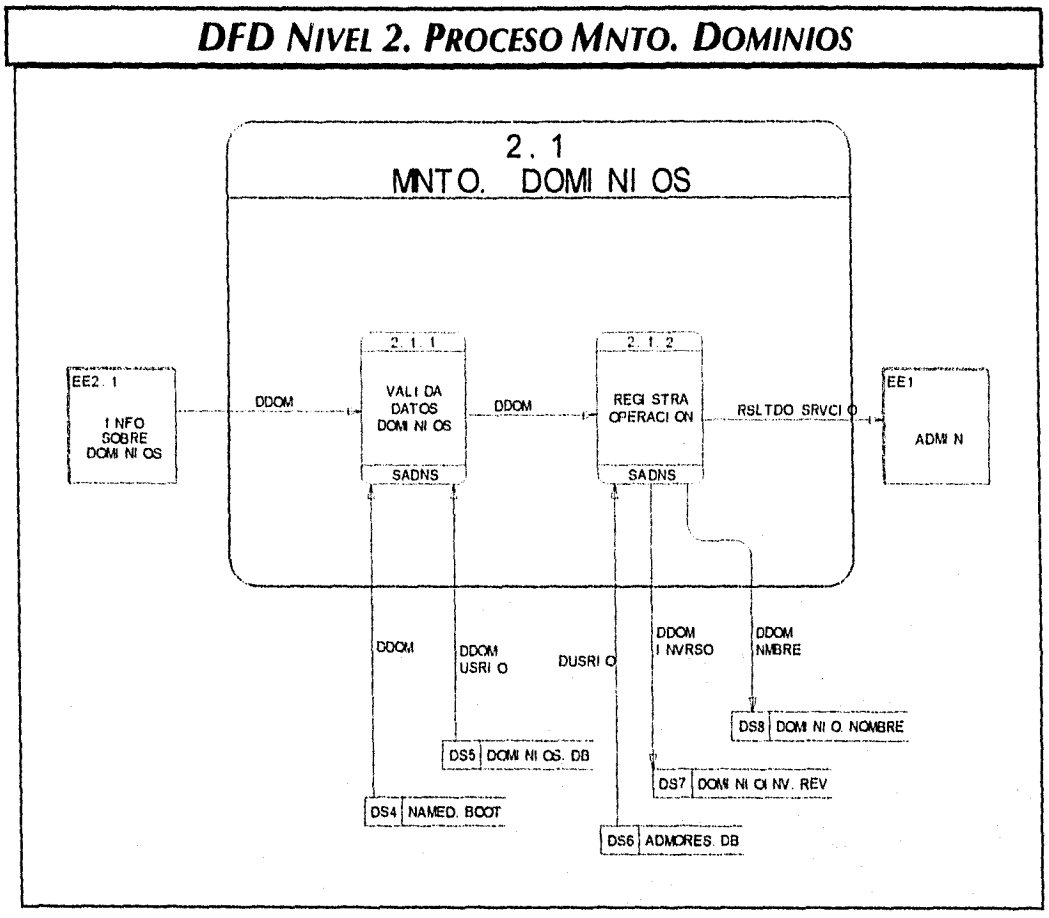


Figura 3-3. DFD del Proceso de Mnto. de Dominios

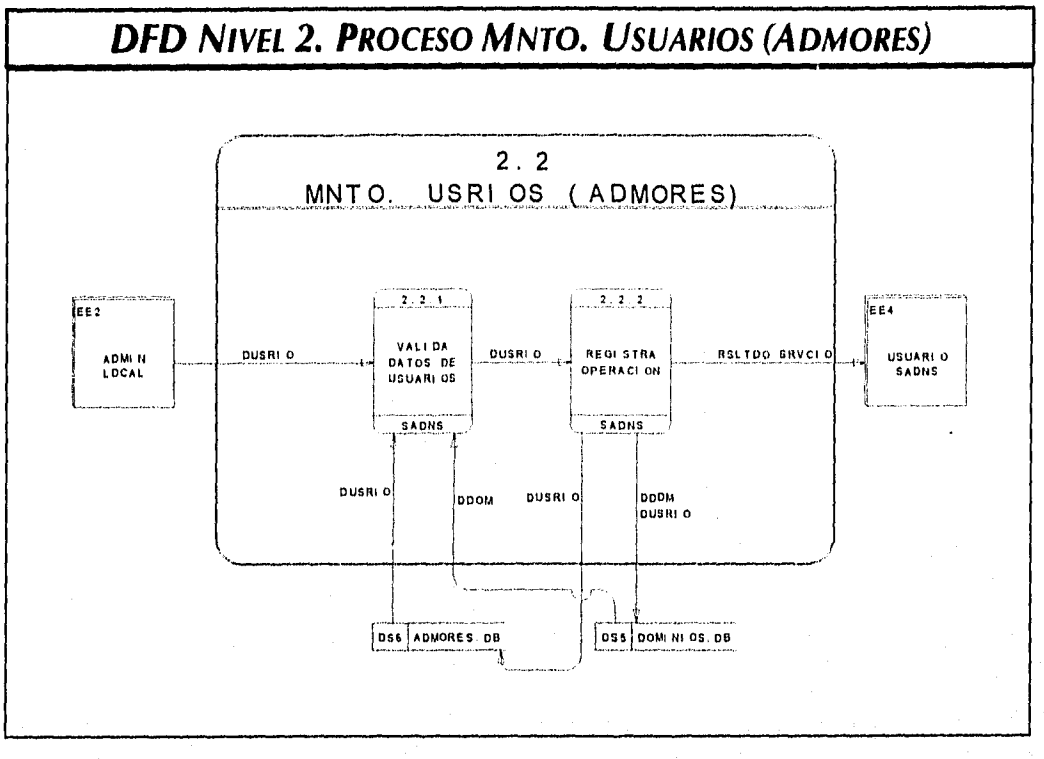


Figura 3-4. DFD del Proceso de Mnto. de Usuarios (Admores)

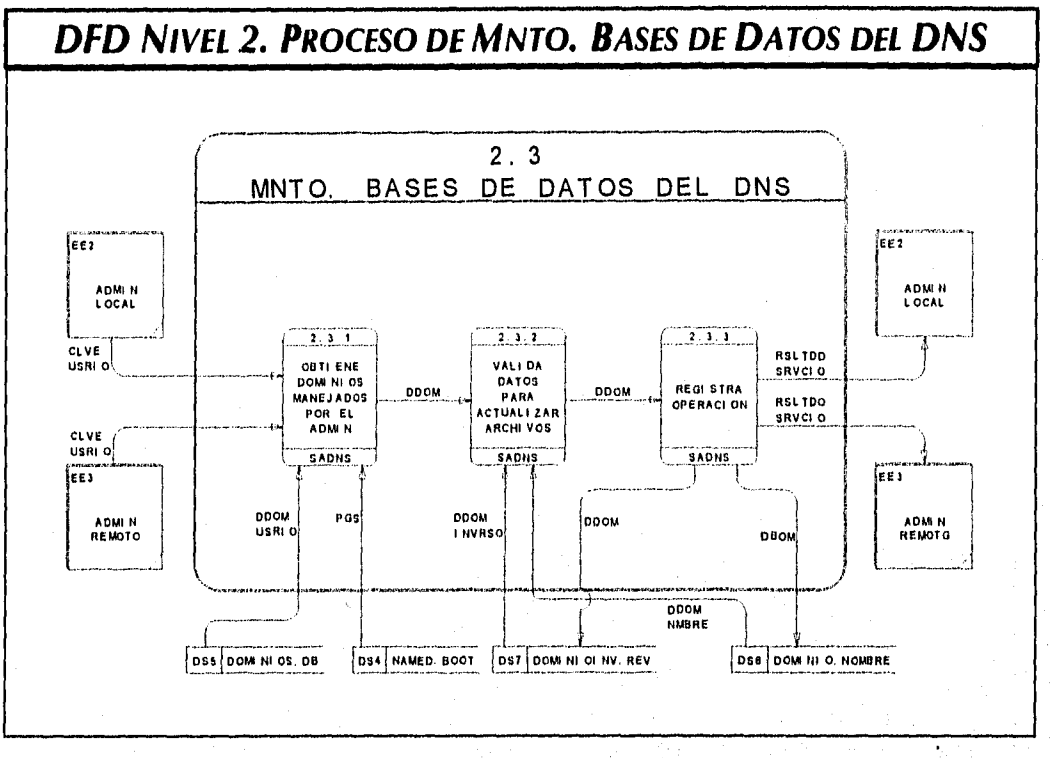


Figura 3-5. DFD del Proceso de Mnto. de Bases de Datos del DNS

3.4 DICCIONARIO DE DATOS

En el análisis de los sistemas, es conveniente tener el diccionario de datos, ya que en él se tiene un conjunto de definiciones de los datos que aparecen en los DFDs (diagramas de datos almacenados o diagramas de flujo de datos). La definición de cada dato se constituye por cada uno de los componentes que forman al dato y las relaciones que existen entre ellos.

El diccionario de datos es muy útil ya que en éste se contemplan todas las definiciones que se manejan en el sistema, es decir, que los datos adquieren una conceptualización de presencia dentro del contexto del análisis. Si uno hace referencia a un dato, uno puede determinar la definición del dato y entender que es lo que se necesita hacer.

En esta sección se muestra el conjunto de estructuras de datos que se manejaron y contemplaron para la realización de este sistema. Así como la colección de elementos que fueron necesarios para analizar y diseñar el sistema. Se puede observar en la figura 3-6, el repositorio de archivos (con sus datos asociados) utilizados para el análisis del sistema.

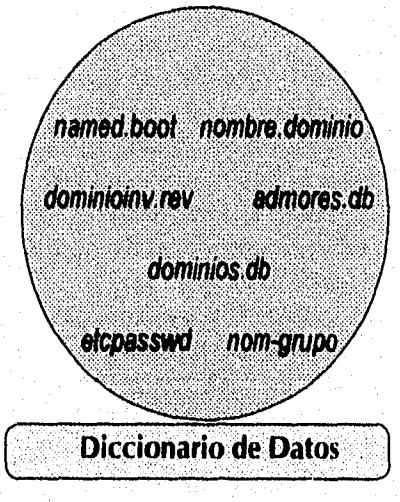


Figura 3-1. Diccionario de Datos (Repositorio de la información)

A continuación se presentan las descripciones de los elementos que intervienen en el análisis del sistema, comenzando con los datos de los DFDs presentados y mostrando los atributos y características de los datos de cada archivo o almacenamiento de datos. También se presenta la descripción de los módulos, datos de control y de intercambio de información de la carta de estructura.

PROCESOS	
Verifica acceso al SADNS	Es un proceso propio del protocolo de httpd que permite la autentificación de usuarios a través de un sistema de archivos. En este procedimiento se validan los usuarios del SADNS (administrador local y remoto).
Mnto. SADNS	Este proceso se refiere al servicio que proporciona el sistema, de administrar y manejar las bases de información del DNS. El servicio se traduce en realizar Altas, Bajas, Cambios y Consultas a las máquinas, dominios y/o procesos.
Mnto. dominios	Es un proceso en el que se definen los servicios de crear, borrar, modificar y consultar dominios dentro del servidor de nombres. Esta tarea de mantenimiento sólo puede ser accesado por el administrador general (root o local).
Mnto. usuarios (Admores)	En este proceso se lleva a cabo los servicios de agregar, eliminar, modificar y consultar usuarios (administradores remotos (UR), los cuales son los que administran uno o más dominios que se encuentran configurados en el servidor de nombres.
Mnto. bases de datos del DNS	Este proceso determina la administración de los servicios de agregar, eliminar, modificar o consultar, los tipos de datos del DNS sobre los archivos de configuración relacionados a cada dominio que administran los usuarios remotos (UR).
Respaldo de archivos y config. y de admon.	Este proceso tiene como fin realizar el respaldo del sistema y las bases de información construida. Se realiza periódicamente o cuando se solicita.
Activación del proceso del DNS	Este proceso efectúa la Inicialización y reactivación del servidor de nombres (in.named o named). Cuando se realiza cualquier cambio en los archivos del DNS, este proceso se tiene que activar para que el servidor de nombres pueda conocer los nuevos datos.
Valida datos dominios	Es un proceso que valida los datos capturados en el sistema tomando en cuenta los tipos de dominios a manejar.
Registra operación	Cuando los datos son validos se toma en cuenta el tipo de acción a realizar y es entonces, cuando este proceso realiza el registro de los datos ingresados.
Obtiene dominios manejados por el admin	Este proceso tiene por objeto obtener los dominios que administran los usuarios del sistema (Local y Remoto). Los dominios están asociados a archivos de configuración conocidas como bases de datos que son mantenidas por los administradores del SADNS.
Valida datos para actualizar archivos	Este proceso tiene como objetivo validar los datos que se introducen para la actualización de archivos de configuración de los dominios.
Valida datos de usuarios	Este procedimiento tiene como función verificar los datos válidos para los usuarios del sistema. Estos datos tiene un orden y regla en la formación de cada campo.

Tabla 3-1. Definición de Procesos

ENTIDADES EXTERNAS	
Admin	Representa al administrador del sistema en cualquiera de sus modos: Administrador Local y Administrador Remoto (root o usuario). Administrador Local o Administrador Root: Es aquel que accesa a todos los módulos del sistema autorizado a agregar, eliminar o modificar dominios dentro de la configuración del DNS y además de poder hacer los mismo para los dominios administrados por los usuarios remotos (externos al sistema local). Administrador Remoto o Usuario: Es el que accesa remotamente, su autoridad es únicamente sobre uno o más dominios, previamente delegados por el administrador root. Administrar a un dominio significa utilizar los archivos de configuración del DNS para poder agregar, eliminar o modificar campos.
Admin Local	Esta entidad define al usuario de nivel más alto, es decir, el administrador local (root). La clave y contraseña es asignada por el propio administrador cuando se instala el sistema.
Admin Remoto	Define a los usuarios del sistema con nivel de autoridad sobre uno o más dominios. Estos usuarios son definidos por el administrador local.
Info Sobre Dominios	En esta entidad se representa a la información sobre los dominios abrindar el mantenimiento.
Usuario SADNS	Son los usuarios (administradores) del sistema que se han actualizado en el SADNS.

Tabla 3-2. Definición de Entidades Externas

FLUJO DE DATOS	
clve usrio	Determina la autorización de acceso al sistema con el esquema de niveles de autoridad (claves de administrador local y administrador remoto).
clve vlda usrio	Es la clave del usuario que ha accesado al sistema, el nivel de autoridad que ha obtenido se define dentro de los módulos del SADNS.
rsltdo srvcio	Proporciona el estado general de los procedimientos. Se pueden obtener reportes, aceptación, rechazo o bien, mensajes de error en la introducción de los datos. Al mencionar srvcio o servicio es interpretando cada módulo de mantenimiento del SADNS.
bndra	Es el valor que se obtiene de falso o verdadero, lo que es aceptación o negación de una acción.
dts SADNS	Son los datos que se necesitan para definir cualquier actividad de servicio que se requiera para el mantenimiento del DNS. Estos datos van desde información de usuarios hasta la información de dominios y tipos de registro del DNS.
ddom	Representa los datos generales de los dominios que permanecen o se integran a los archivos de configuración o bases de datos del DNS.
dusrio	Son los datos del usuario que se agregan, eliminan, modifican o consultan a las bases de configuración del sistema. En estos datos se incorporan los grupos a los que pertenecen los usuarios, es decir, administrador local (general,root) y administrador remoto (de sólo dominios). También se indican los datos del usuario que realiza acciones de mantenimiento.
ddom usrio	Son los datos de uno o más dominios que están en el servidor de nombres. Se incorpora información de los usuarios que administran a dichos dominios. Describe el o los dominios que los usuarios (local o remoto) administran. Establece la autoridad que tienen los administradores sobre los dominios.
ddom invrso	Son estructuras de datos relacionadas al tipo de registro del DNS que vinculan a los dominios inversos.
archivos del SADNS	Son el conjunto de archivos de configuración (bases de datos o bases de información) y programas que forman al SADNS.
nmro prcso	Es el número de proceso (pid) del servidor de nombres dentro del sistema operativo. Este sirve para identificar al proceso y poder reinicializarlo.
ddom nmbre	Son datos relacionados a tipos de registros del DNS vinculados a los archivos de dominios por nombre.
ddom dusrio	Son los datos de los dominios y claves de los usuarios que administran a ellos. En este flujo se pretende transmitir la información relacionada a los usuarios en cuanto a su manejo de dominios.
pos	Define la posición de las bases de datos dentro del sistema de archivos. La posición se define dentro del archivo named.boot.

Tabla 3-3. Definición de Flujo de Datos

Archivos pertenecientes al sistema, definidos dentro del Diccionario de Datos:

ALMACENAMIENTO DE DATOS o ARCHIVOS	
named.boot	Archivo de configuración del DNS. Ahí se define el lugar donde las bases de datos permanecerán y además se definen los tipos de servidores que estarán configurados (primario, secundario).
nombre.dominio	Este archivo mantiene la información sobre un dominio por nombre.
dominioinv.rev	Este archivo mantiene la información sobre un dominio inverso (por número de red).
dominios.db	En esta base se almacenan los administradores de los dominios existentes en el servidor de nombres.
admores.db	Se almacena la información relacionada a los administradores de red que manipularán su administración distribuida de uno o más dominios.
etcpasswd	Es el archivo donde se guarda la clave del usuario junto con su contraseña encriptada.
nom-grupo	Es el archivo donde se definen los grupos de usuarios que pueden accesar al sistema.

Tabla 3-4. Definición de Bases de Datos

A continuación se presentan los elementos que forman a cada uno de los archivos definidos en el diccionario de datos. Ver tabla 3-5.

ARCHIVO	DESCRIPCION DEL REGISTRO QUE LO FORMAN
	directory+trayectoria (configurable una sola vez)
	cache+. (configurable una sola vez)
named.boot	primary+dominio+nombre-bd*
	secondary+dominio+dirip-serv-prim+copia-bd
	**Donde bd=base-de-datos y cada renglón es un registro*
nombre.dominio	nombre +clase+tipo-registro+dato
dominioinv.rev	nombre+clase+tipo-registro+dato
dominios.db	dominio+clave+tipo-dominio+fecha
admores.db	clave+nom-grupo+contraseña+dependencia+dirección+
	nombre-usua+ puesto+correo+tel+fax.
etcpasswd	clave+contraseña
nom-grupo	nom-grupo+clave

Tabla 3-5. Elementos de los Archivos del SADNS

A continuación se define cada elemento que forma las estructuras de datos de los archivos.

ARCHIVO	ELEMENTO	DEFINICION
	primary	Indica que se configura un servidor de nombres primario.
	dominio	Puede ser de nombre o numérico (inverso).
named.boot	*bd*	Es la base de datos que almacenará todos los tipos
	secondary	Indica que se configura un servidor de nombres primario
	dirip-serv-prim	Es la dirección de Internet IP del servidor primario.
	copia-bd	Es una copia de la información del servidor primario.
	nombre	Define un valor relacionado al tipo de registro del DNS.
	clase	La clase define el tipo de red . IN de Internet.
nombre.dominio	*tipo-registro*[5]	El tipo de registro en este archivo, representan a aquellos que se relacionan con los nombres lógicos.
	dato	Es el valor que especifica la identidad del nombre.
	nombre	Define un valor relacionado al tipo de registro asociado a a este archivo.
dominioinv.rev	*clase*	La clase define el tipo de red . IN de Internet
	*tipo-registro**	Se presentan tipos de registro relacionados al archivo.
	dato	Es el valor real que se asigna a cada tipo de registro.
	dominio	El dominio manejado por algún administrador de red.
dominios.db	*clave*	Es la clave del administrador de red.
	tipo-dominio	Es para definir el tipo de dominio (nombre, inverso).
	fecha	La fecha cuando se delegó la administración del dominio.
	clave	Es la clave de acceso del administrador.
	nom-grupo	Es el grupo a que pertenece el administrador.
	contraseña	Es un valor encriptado que valida la autentificación.
	dependencia	Es la definición del lugar que representa al administrador.
admores.db	*dirección*	Es la dirección física de la dependencia.
	nombre-usua	Es el nombre personal del administrador de red.
	puesto	Define la ocupación dentro de la dependencia.
	correo	Es la dirección electrónica del administrador para envíos.
	tel	Representa el teléfono del administrador.
	fax	Se define el número de fax de la dependencia.
etcpasswd	*clave*	Es el identificador para acceso al sistema.
	contraseña	Es una palabra encriptada para verificar la autentificación.
nom-grupo	*nom-grupo*	Define el nombre del grupo (dependiendo de la autoridad).
	clave	Es el identificador para acceso al sistema.

Tabla 3-6. Definición de Elementos en Archivos del SADNS

[5] *tipo de registro* depende del archivo y de la información que se está manipulando dentro de la configuración del DNS.

En las tablas anteriores, se han presentado las definiciones de las bases de datos, así de los elementos de cada registro que intervienen en el sistema. A continuación se presentarán los registros

El diccionario de datos queda completo cuando se define cada estructura que está formada en el archivo. Cabe hacer notar que en el archivo *named.boot* se mantiene la definición de la forma en que se encuentra configurado un servidor de nombres y además la posición de cada archivo físico en un servidor determinado. Este archivo está formado por registros múltiples, es decir que no tienen un comportamiento similar, por lo que en el modelado de datos se tratará de presentar como si fueran diferentes bases de información.

Características de Los Campos de Registros de Cada Archivo de Parámetros

A continuación se presenta como está formado cada registro y las características de los elementos de cada uno de ellos dentro de los archivos de configuración definidos como bases de información o archivos de parámetros. Los campos que representaremos son: campo, longitud y tipo. Donde campo, representa un elemento del registro de la base de datos. La longitud me indica el tamaño que ser reserva para el dato. El tipo de dato es para determinar que clase de dato se manipulará en el sistema, tal como, caracter, númerico y alfanumérico.

named.boot

Dentro de este archivo de parámetros hay la existencia de valores de parámetros los cuales se definen en estos diferentes bloques. Tales como *directory, cache, primary, secondary*. El valor del parámetro o campo esta definido en los bloques.

CAMPO	TIPO DATO	LONGITUD
directory	9	caracter
dir-dato	20	alfanumérico

CAMPO	LONGITUD	TIPO DATO
cache	9	caracter
.	1	caracter
archivo-root	20	alfanumérico

CAMPO	LONGITUD	TIPO DATO
primary	7	caracter
dominio	15	alfanumérico
nombre-db	40	alfanumérico

Tabla 3-7. Elementos de *named.boot*

CAMPO	LONGITUD	TIPO-DATO
secondary	7	caracter
dominio	15	alfanumérico
dirip-ser	30	alfanumérico
nombre-db	40	alfanumérico

Tabla 3-8. Continuación de elementos de *named.boot*

Donde los valores definidos son: ***directory, cache, primary, secondary.***

nombre.dominio

En este archivo de configuración, existe una forma de agregar los datos de una forma ordenada y sistemática. No todos los tipos de registro son agregadas en este tipo de archivo, sólo algunos tipos que tienen relación con los dominios por nombre.

CAMPO	LONGITUD	TIPO DATO
nombre	30	alfanumérico
clase	2	caracter
*tipo-registro	5	caracter
dato	40	alfanumérico

Tabla 3-9. Elementos de *nombre.dominio*

*tipo-registro: Puede tomar de valor a SOA, NS, A, CNAME, MX.

dominioinv.rev

En este archivo de configuración intervienen solamente los datos que intervienen en los dominios inversos (de dirección inversa).

CAMPO	LONGITUD	TIPO DATO
nombre	30	alfanumérico
clase	2	caracter
**tipo-registro	3	caracter
dato	40	alfanumérico

Tabla 3-10. Elementos de *dominioinv.rev*

**tipo-registro: Unicamente para los datos SOA, NS, PTR.

dominios.db

En este archivo de parámetros se almacenan las claves de los administradores y los dominios manejados. Es un archivo tipo tabla.

CAMPO	LONGITUD	TIPO DATO
dominio	15	alfanumérico
clave	8	alfanumérico
tipo-dominio	7	caracter
fecha	9	alfanumérico

Tabla 3-11. Elementos de *dominios.db*

admores.db

En este archivo se guardan los datos en forma de una tabla plana. Los datos se agrupan por renglón y separados por ":" (dos puntos).

CAMPO	LONGITUD	TIPO DATO
clave	8	alfanumérico
nom-grupo	8	caracter
contraseña	8	caracter
dependencia	50	alfanumérico
dirección	50	alfanumérico
nom-usuario	50	caracter
puesto	50	caracter
correo	40	alfanumérico
tel	10	alfanumérico
fax	10	alfanumérico

Tabla 3-12. Elementos de *admores.db*

etcpasswd

Archivo plano, cuyo campo de la contraseña es encriptada. Este archivo tiene registros de dos campos.

CAMPO	LONGITUD	TIPO DATO
clave	8	alfanumérico
contraseña	8	alfanumérico

Tabla 3-13. Elementos de *etcpasswd*

nom-grupo

Este archivo es un archivo de configuración genérica, para el funcionamiento de la autentificación.

CAMPO	LONGITUD	TIPO DATO
nom-grupo	8	alfanumérico
contraseña	8	alfanumérico

Tabla 3-14. Elementos de *nom-grupo*

Finalmente se tiene una completa visualización de los datos (con sus características respectivas), así como la posibilidad de definir en cualquier momento un determinado campo o registro dentro del sistema.

5.5 MODELADO DE DATOS

El modelado de datos es representar el significado de los datos. Generalmente en los modelos de datos se representan campos de los registros, atributos y relaciones.

Se puede mencionar que un campo de un registro es un elemento de la vida real, un concepto o evento. Un atributo es una propiedad o característica que distingue al campo. Una relación es la asociación lógica entre los campos y los archivos (bases de datos).

FORMACIÓN DE ARCHIVOS

Primeramente definiremos la formación de los archivos que están implicados en el sistema: named.boot, nombre.dominio, dominioinv.rev, dominios.db, admores.db, etcpasswd, grupo.

named.boot *(Como mencionamos anteriormente, éste está formado por diferentes registros).*
Formaciones de las bases, representación de los registros.

DIRECTORIO	DIR DATO
directory	/var/named

*1)

RAIZ	ARCHIVO ROOT
cache	named.ca

*1)

TIPO SERVIDOR	DOMINIO	NOMBRE BD
primary	0.0.127.in-addr.arpa	named.local
primary	unam.mx	unam.mx
primary	sar.net	sar.net

**2)

TIPO SERVIDOR	DOMINIO	DIRIP SERV PRIM	COPIA BD
secondary	248.132.in-addr.arpa	132.248.204.1	248.132.rev.bak
secondary	unam.mx	132.248.204.1	unam.mx.bak
secondary	sar.net	200.13.64.1	sar.net.bak
secondary	64.13.200.in-addr.arpa	200.13.64.1	64.13.200.rev.bak

**2) Tabla 3-15. Formación de *named.boot*

NOTA:
*1) Una sola definición dentro del archivo named.boot.
**2) Más de una definición en el archivo named.boot.

nombre.dominio (De esta tabla pueden existir varias archivos similares, únicamente cambia el nombre de dominio).

Cuando se forma una base de esta estructura varía para cada dominio que se crea. Por ejemplo para *sar.net*.

NOMBRE	CLASE	TIPO-REGISTRO	DATO
@	in	soa	dns.sar.net. buzon.sar.net. (serial refresh retry expire minimun)
	in	ns	dns.sar.net.
chajul	in	a	200.13.64.1
ftp	in	cname	chajul
sar.net.	in	mx	10 chajul.sar.net.

Tabla 3-16. Formación de *nombre.dominio*

Se debe señalar que en este tipo de archivos, los tipos-registro pueden haber más de uno en la tabla excepto el tipo de registro *soa*.

dominioinv.rev (Pueden surgir tablas similares a este para cada dominio inverso).

En este archivo se registran los registros de dominio inverso. Se genera una tabla por cada dirección de red delegada que tenga el administrador.

NOMBRE	CLASE	TIPO REGISTRO	DATO
@	in	soa	dns.sar.net buzon.sar.net (serial refresh retry expire minimun)
	in	ns	dns.sar.net.
1	in	ptr	chajul.sar.net.
2	in	ptr	solar.sar.net.

Tabla 3-17. Formación de *dominioinv.rev*

Se puede observar que en este tipo de tabla para representar el dominio inverso, se utilizan únicamente los registros soa, ns y ptr.

NOTA: Los tipos de registros que se utilizarán en los archivos nombre.dominio y dominioinv.rev del sistema son: SOA, NS, A, MX, PTR.

dominios.db (Relacionan a los administradores con los dominios en el sistema).

DOMINIO	CLAVE	TIPO-DOMINIO	FECHA
sar.net	mdeleo	nombres	13-ene-96
64.13.200.in-addr.arpa	mdeleo	inverso	13-ene-96
syscase.com.mx	noble	nombres	16-abr -96

Tabla 3-18. Formación de *dominios.db*

admores.db (Tabla de representación de los administradores del sistema).

CLAVE	NOM-GRUPO	CONTRASEÑA	DEPENDENCIA	DIRECCION
mdeleo	admin	KL/-s33.	SOLAR	Camino Real 60
noble	admin	LPz&\K	SYSCASE	Div. del Norte 103
armando	admin-root	:LMXz8&	SOLAR	Camino Real 60
NOM-USUARIO	**PUESTO**	**CORREO**	**TEL**	**FAX**
Michael de Leo	Director Tec	mdeleo@sar.net	420-5900	620-5909
Sergio Noble	Director Gral	noble@unam.mx	679-4900	679-4901
Armando A. A.	Gte. del NOC	armando@sar.net	420-5900	420-5909

Tabla 3-19. Formación de *admores.db*

Los siguientes archivos etcpasswd y grupo son para que se pueda autentificar el usuario que accesa al sistema.

etcpasswd (Mantiene la tabla de password del sistema de acceso)

CLAVE	CONTRASEÑA
armando	:LMXz8&
mdeleo	KL/-s33.
noble	LPz&\K

Tabla 3-20. Formación de *etcpasswd*

nom-grupo (Se definen los usuarios para cada grupo al que pertenecen)

NOM-GRUPO	CLAVE
admin-root	armando
admin	mdeleo
admin	noble

Tabla 3-21. Formación de nom-grupo

3.6 MODELO DE IDENTIDAD RELACIÓN

Finalmente, se puede observar el modelo de entidad relación que representa a las bases de datos (archivos de parámetros). Ver el siguiente diagrama.

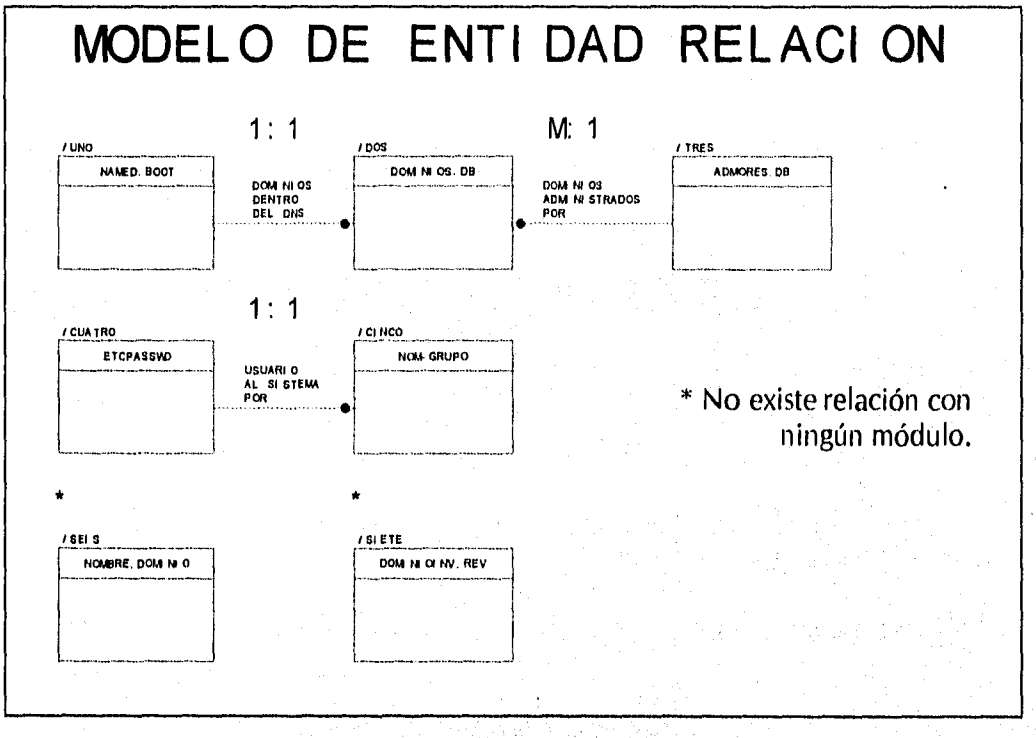


Figura 3-7. Modelo de Entidad Relación del SADNS

A continuación se presentan los elementos que conforman a cada base de datos utilizada en el SADNS con la finalidad de visualizar correctamente parte del modelado del sistema. Ver diagrama siguiente.

Se muestran las representaciones de los archivos named.boot, dominios.db, admores.db, nombre.dominio, domininv.rev, etcpasswd y nom-grupo.

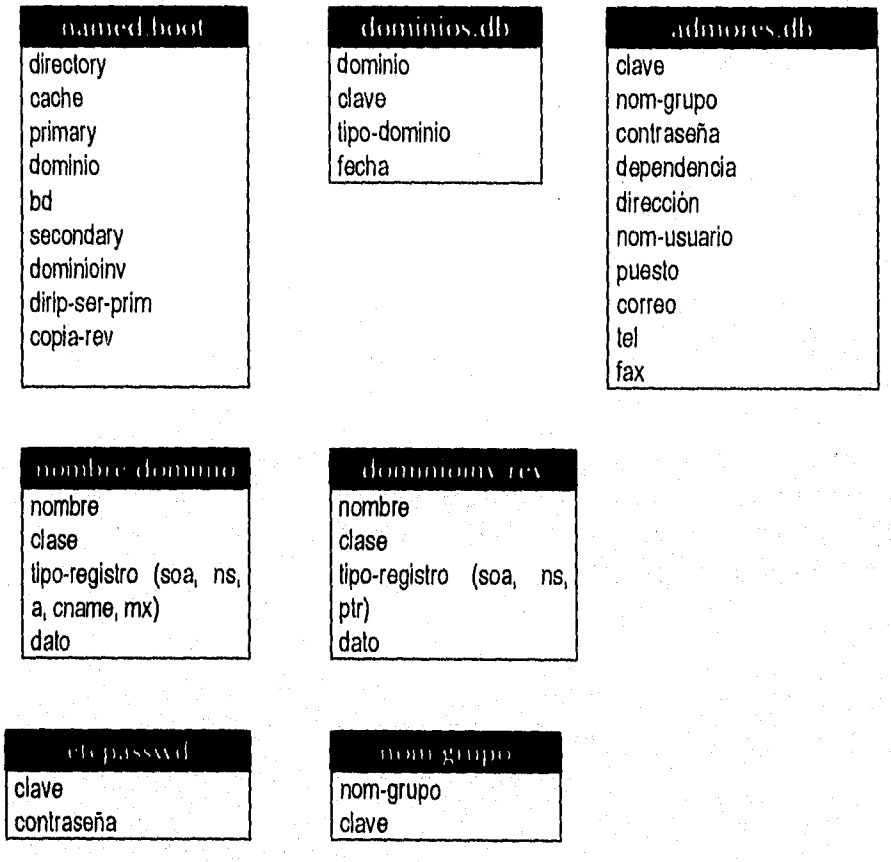


Figura 3-8. Atributos de las Bases de Información del SADNS

Los diagramas presentados en esta parte muestran los elementos que forman a cada uno de los archivos de parámetros que intervienen en el sistema presentado.

3.7 NORMALIZACIÓN

Al realizar el modelado de los datos, nos pudimos dar cuenta de que no es necesario realizar la normalización de datos, debido a que no estamos utilizando las bases de datos tradicionales (indexadas), sino, que utilizamos puramente archivos de parámetros y de configuración en texto.

Por lo que dado esto, no es necesario normalizar los archivos de texto de parámetros que intervienen en el sistema, o bien, como se conoce en ambientes Unix, bases de datos convencionales de texto plano.

El DNS trabaja con un programa ejecutable en C que manipula los archivos de configuración o arranque y conoce donde están los sitios destinado para devolver información.

La convención de manipular archivos de texto, es debido a que la portabilidad de estos archivos son de gran utilidad, además por no tener un gran número de datos por utilizar no es necesario tener un manejador de bases de datos tradicionales (datos indexados).

3.8 DIAGRAMA ESTRUCTURADO

El diagrama estructurado o carta de estructura del sistema es la representación de la información (como fluye) dentro del sistema (SADNS). Es importante mencionar que se obtuvo el diagrama estructurado en base a los procesos realizados en los diagramas de flujo de datos (DFDs) debido a que son los procedimientos lógicos del sistema y la carta es como están jerárquicamente los programas funcionando e interactuando con los servicios que proporciona el sistema.

A continuación se describen los módulos (programas, en nuestro caso CGIs) que intervienen en la carta estructurada.

DESCRIPCIÓN DE MÓDULOS DE LA CARTA DE ESTRUCTURA PROGRAMAS DEL SADNS	
HTTPD	Programa del servidor de Web que brinda la interface y la autorización
MENU PRINCIPAL	Presenta las opciones de cada administrador
ADMINDOM.HTML	Realiza el mantenimiento de dominios
ADMINUSU.HTML	Realiza el mantenimiento de los administradores (usuarios loca y remoto)
ABCMAQ.AWK	Realiza el mantenimiento de las bases de datos. En el proceso de implantación este módulo adopta el nombre de DBASES
RESPALDO DE ARCHIVOS	Programa en realiza el respaldo de los archivos
ACTIVAPR.SH	Activa y reinicializa el servidor de nombres named
HERRAMIENTAS.HTML	Programa implantado para proporcionar herramientas
CREA.HTML	Programa que genera archivos de configuración del DNS
DIG.HTML	Brinda posibilidad de hacer uso del programa dig
DOC.HTML	Permite utilizar el programa doc
NSLOOKUP.HTML	Módulo que permite utilizar el programa nslookup
USUARIOS.HTML	Ejecuta el alta de usuarios (administradores) al SADNS
BAJA.SH	Realiza la baja de usuarios del SADNS
CAMUSU.SH	Módulo que permite el cambio de usuarios en SADNS
CONSULTA.SH	Programa que permite consultar a usuarios
DOMINIOS.HTML	Realiza el alta de dominios al DNS
BORRA.AWK	Borra dominios del DNS
CAMBIA.AWK	Cambia dominios del DNS
DNS.AWK	Realiza la consulta de los dominios del DNS

Tabla 3-22. Descripción de Módulos del Diagrama Estructurado del SADNS

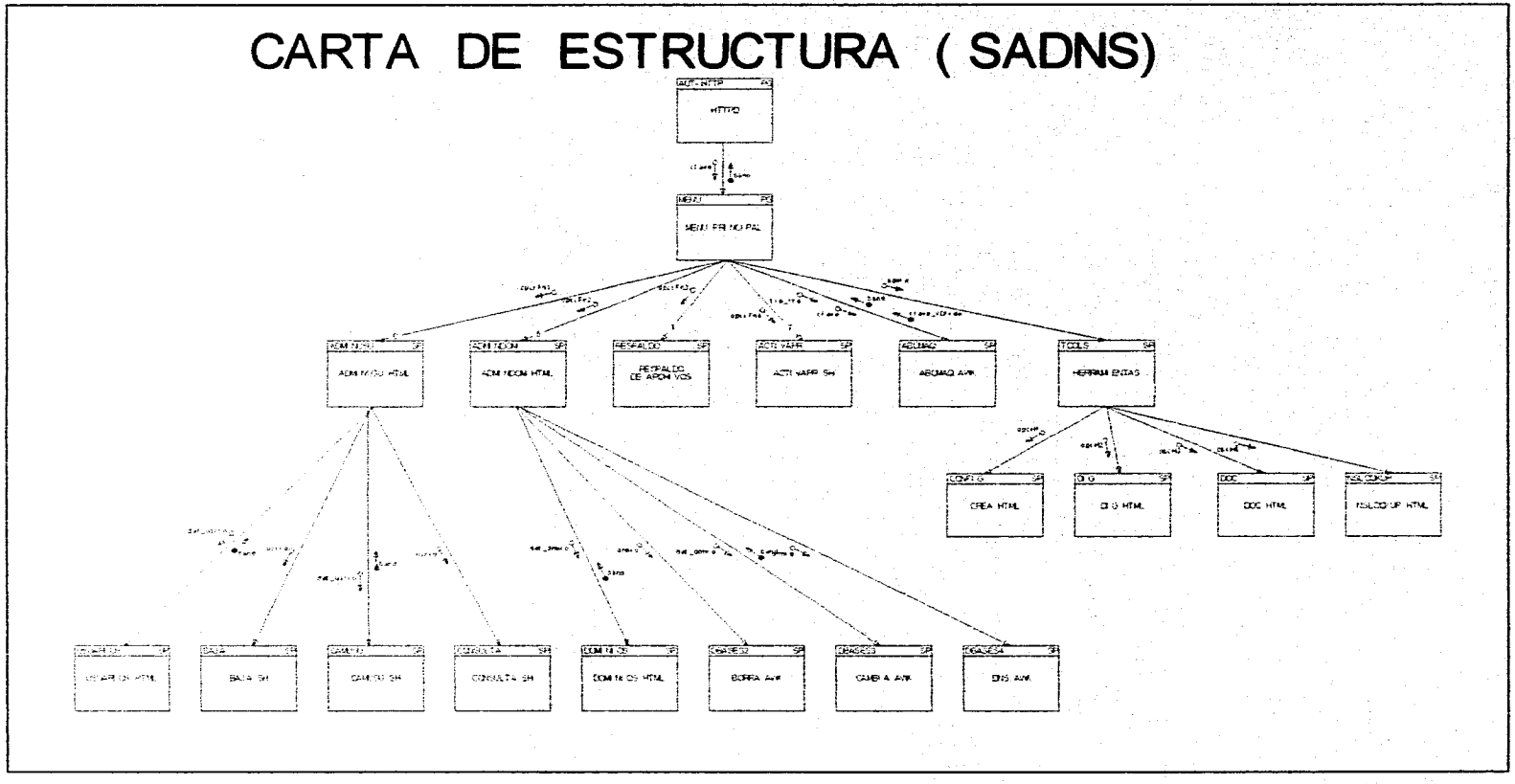


Figura 3-9. Diagrama Estructurado o Carta Estructurada del SADNS

4. Desarrollo del Sistema

4.1 Introducción al Desarrollo

El Sistema de Automatización del DNS es un proyecto que surgió como una necesidad para administrar en forma distribuida un conjunto de información que permanece sobre un mismo servidor.

Para desarrollar este sistema se tomaron en cuenta las siguientes características de uso:

- acceso remoto,
- interface gráfica fácil de manejar,
- modos de administración (administración local y remota),
- administración distribuida de los dominios por cada usuario,
- autoridad y permisos designados por el administrador principal (administrador local) y
- ayuda e información del DNS en línea,
- posibilidad de respaldar y utilizar la información de los archivos generados por cada dominio.

Partiendo de estas características, se determinó desarrollar el sistema sobre un protocolo distribuido en Internet como lo es el servidor de World Wide Web (WWW) o Web, conocido como Hyper Text Transfer Protocol Daemon (HTTPD). Más adelante se comentarán las características que describen al servidor de *httpd*.

El servidor de Web utiliza un lenguaje de programación HTML conocido como Hyper Text Markup Language (descrito más adelante). Se decidió desarrollar en el lenguaje HTML 3.0 y con extensiones para Netscape ®. Además que este servidor tiene como clientes o navegadores de sus recursos a programas que son interfaces gráficas fáciles de configurar y utilizar. Por lo que también se decidió por convención y uso general el utilizar el navegador de Netscape ® versión 1.0 o mayor.

4.2 DETERMINACIÓN DE LA PLATAFORMA DE SOFTWARE

El World Wide Web (WWW) es un sistema de información de hipertexto gráfico, distribuido, multiplataforma, dinámico, interactivo y global que funciona sobre Internet.

El servidor de WWW es un programa que está sobre una máquina de la red, esperando ser accesado por algún cliente y hacer una petición, devolviendo generalmente algún archivo. El WWW funciona con el modelo cliente-servidor. El cliente es conocido como navegador o *browser* y el servidor es el *httpd* (hiper text transfer protocol daemon) o demonio de Web. Tanto servidores como clientes se comunican e interactúan utilizando el *httpd*, que es un lenguaje especial diseñado para transferir documentos basados en hipertexto que estén sobre la red. Los servidores estándares de Web, permiten manejar:

- Lenguaje HTML (1.0, 2.0 o 3.0 con extensiones especiales de algunos servidores)
  - Manejo de Imagenes
  - Manejo de Sonido
  - Tablas
  - Formas de captura (método GET y POST)
- CGI (Common Gateway Interface)

El HiperText Markup Language (HTML) es un lenguaje de programación orientado a hipertexto a través de etiquetas o marcas que permiten obtener ciertos resultados. Es una derivación del Standard Generalized Markup Language (SGML) que permite la descripción de la estructura general de diferentes tipos de documentos. Generalmente los archivos que se crean con este HTML tienen la extensión .html (con ella el servidor reconoce el tipo). Con el HTML se pueden crear documentos estructurados con diversas características como son:

- Inicio del documento
- Encabezados
- Cuerpo del documento
  - Listas, párrafos, tipografía y formateo del documento.
- Fin del cuerpo
- Fin del documento

Se presenta a continuación la estructura general de un documento en HTML (ver figura 4-1) :

```
<HTML>
<HEAD>
<TITLE>Mi primer hoja de WWW
</TITLE>
</HEAD>
<BODY>
<H1>Este es mi primer documento en HTML</H1>
</BODY>
</HTML>
```

Figura 4-1. Ejemplo de un Archivo en HTML.

<HTML> Es la primera etiqueta en cualquier documento HTML, este indica que el contenido del archivo es en el lenguaje HTML.
<HEAD> Esta etiqueta indica que las líneas dentro de los puntos de inicio y final son los que se prolongan a el resto del archivo.
<TITLE> Son las marcas que especifican el título del documento.
<BODY> Representa que es el cuerpo del documento formado por listas, encabezados con diferentes tamaños de tipografía, imágenes, ligas, formas, etc.

La representación de </ > es cuando se determina la finalización de la etiqueta en cuestión.

Dentro del HTML existe la posibilidad de trabajar los datos interactivamente, esto se puede realizar a través de la implantación del llenado de formas y posteriormente estos datos que se capturan se puede enviar a través de una liga a un Uniform Resource Location (URL), donde este URL puede ser una liga a un documento dentro del mismo servidor, otro servidor de Web a través de la red, un protocolo dentro de Internet (gopher, telnet, ftp, wais, etc) o bien, un CGI que procese la información y devuelva resultados al servidor de Web, para que esté lo despliegue al cliente o navegador.

Una forma en el Web es la captura de los datos a través de una interface amigable y fácil de utilizar, mediante, campos de captura mediante el teclado, cajas de diálogo a través de elecciones de barras o elección múltiple de datos, todos presentados gráficamente. Mediante etiquetas usando HTML se puede representar lo que uno desea capturar como si fuera un sistema tradicional de capturación, la ventaja de utilizar al servidor de Web es que estas estructuras ya están interconstruídas con el protocolo. La única manera de que las capturas se procesen es utilizando un *CGI (Common Gateway Interface).*

Un CGI o *script* es una interface que proporciona una comunicación entre una página interactiva (forma) hacia un programa ejecutable escrito en cualquier lenguaje de programación que soporte ciertos requisitos, y devuelva lo procesado al cliente que solicitó el procesamiento de los datos. Los CGIs pueden ser programas hechos en el lenguaje C, perl, awk, nawk, sh, csh, manejadores de base de datos como informix, oracle, etc.

Teniendo en cuenta que, inicialmente el servidor httpd o bien servidor de Web comenzó su popularidad en ambientes UNIX y también sus nuevas extensiones primeramente se presentan en este sistema operativo, además que los servidores Web aprovechan las herramientas y conceptos que los sistemas UNIX ofrecen para demostrar sus habilidades y características, se determinó que el Sistema de Automatización del DNS (SADNS) se desarrollaría sobre una estación UNIX, pudiendo se utilizar en arquitecturas de UNIX como SUN, SGI, Linux (PC), Ultrix, todos basados en UNIX BSD (Berkeley Software Distribution).

Dado que el sistema estaría sobre un sistema operativo UNIX y con un servidor de Web usando a un CGI, se determinó que para tener una amplia capacidad de soportar la plataforma de software en diferentes plataformas de hardware, se utilizarían el interprete del sistema operativo *shell* (sh), utilerías de UNIX (awk,nawk, grep) y un programa en C para encriptamiento de passwords.

4.3 DESARROLLO DEL SADNS

Hemos mencionado los requerimientos de software que se necesitaron para poder desarrollar este sistema. Realmente, las utilerías de UNIX vienen con el sistema operativo incluido y proporcionan su manual en línea, por lo que es más fácil programar sobre el servidor UNIX.

Finalmente, vamos a mencionar el esquema de requerimientos que se utilizó para implantar el SADNS:

- Sobre un Servidor UNIX
- Sobre un Servidor Web
- Usando un CGI (Programa ejecutable hecho en sh, nawk y C)
- Navegador o browser de Web (Netscape ® versión 1.0 o mayor)
- Acceso a Internet

Debido a que uno de las metas del sistema perseguía que el SADNS corriera sobre plataformas UNIX bajo diferente arquitectura de servidor (SUN, SGI, DEC, Linux PC) se eligió un servidor UNIX de Silicon Graphics Inc. (SGI) con sistema operativo IRIX 5.3 (Con System V y BSD) para implantar el sistema SADNS.

El programa *httpd* utlizado se obtuvo de *httpd* de National Center for Supercomputing Applications (NCSA) versión 1.5c que maneja extensiones estándares de HTML. También existen otros servidores como los de CERN, el de WINWEB.EXE (para PC, que no soporta CGI), servidor de Spry etc.

Debido a la popularidad del servidor httpd de NCSA, se configuró en la SGI (El archivo fuente está en el servidor de ftp de NCSA localizado con el siguiente URL: *ftp://ftp.ncsa.uiuc.edu/pub/Web/httpd/Unix/ncsa_httpd/httpd.tar.Z*) para poder comenzar a programar el CGI. Para la autentificación de usuarios en el Web, se recurrió a utilizar lo denominado Interface de Autentificación de usuarios (configuración y detalles en el Anexo C), con la finalidad de que los administradores que quisieran entrar al servidor de Web fueran los que se dieron de alta previamente cpm autorización por el administrador principal.

Para la realización del CGI se consideró que los datos capturados se procesarían sobre un conjunto de módulos que agregaran, borraran o modificaran tablas en UNIX. En este sistema operativo, todas las tablas pasan a ser bases de datos del sistema, por eso para la programación del CGI se optó por una utilería de UNIX básica que todo sistema operativo UNIX debe traer en su sistema. Una

utilería de UNIX es considerada como un comando útil, integrado y capaz de obtener resultados rápidamente dentro del sistema operativo Unix.

El programa o comando que se utilizó en este proyecto para integrar al Web con los procesamientos de la información fue el awk (versión más actualizada nawk) que es uno de los comandos más complejos del Unix en general, sobre todo porque no es tan solo un comando que realiza una sola tarea, sino que es un lenguaje de programación bien construido, que proporciona facilidades para el manejo de archivos planos, es decir, para descomposición de líneas, reconocimiento de patrones (expresiones regulares), cálculos aritméticos, manejo de funciones , arreglos de cadenas, control de flujo y en general es muy similar al lenguaje C.

awk fue escrito por Alfred Aho, Peter Weinberger y Brian Kernighan en el año de 1997. El nombre de este lenguaje fue tomado de las iniciales de sus autores, a diferencia de otros comandos en Unix que son acrónimos de la función que realiza el comando. A medida que el tiempo avanzaba, se escribieron otras versiones de awk con mayores habilidades, es el caso de nawk que es el que se utilizó en la programación de este sistema.

Finalmente, para poder accesar al SADNS, tiene que estar cualquier cliente (administrador de red) con algún programa navegador de Web, para poder ingresar al sistema de administración distribuida. Como es un sistema que funciona para configurar y mantener un servicio de Internet, el SADNS se pone a disposición de todos los usuarios que administran dominios que tengan acceso a Internet a través de la UNAM o cualquier otro proveedor del servicio.

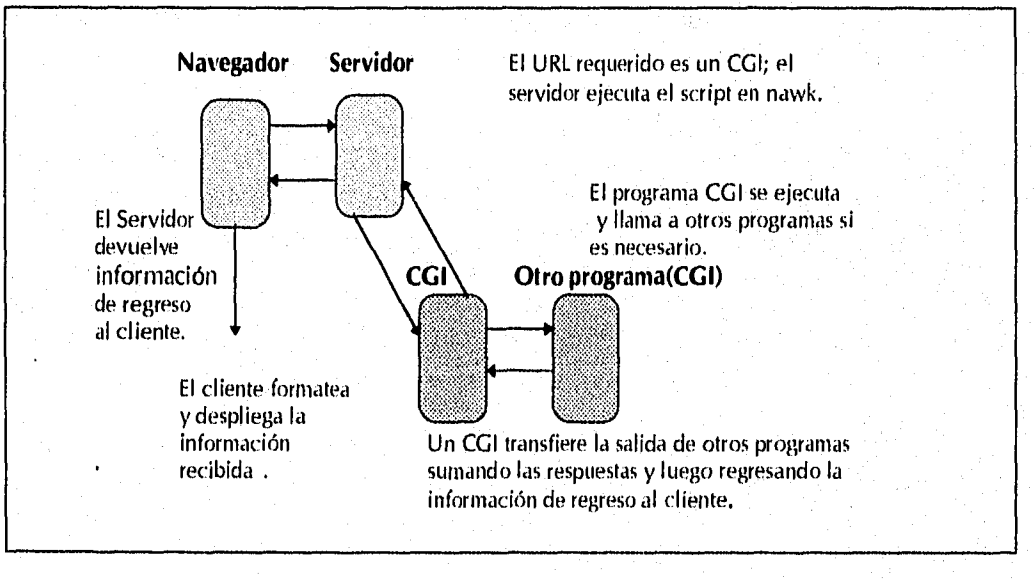


Figura 4-2. Funcionamiento del Cliente-Servidor de Web Interactuando con un Programa

4.4 INSTALACIÓN

Para llevar a cabo la instalación del sistema se requiere de cierta configuración previa en la máquina UNIX. A continuación se dan los detalles de los requerimientos que se requieren para tener utilizando el SADNS.

1. Configurar un servidor de nombres primario
2. Tener corriendo un servidor de Web, más las configuraciones de autentificación
3. Colocar el CGI, correspondiente al programa que intercomunica al sistema con los servidores involucrados (DNS y *httpd*)
4. Tener espacio suficiente en disco para soportar la información de los dominios primarios, se requiere mínimamente 5 Mb.

Con los pasos anteriores de instalación se tiene completamente configurado el Sistema de Automatización del DNS. Los detalles de la configuración se presentan en el Anexo C.

A pesar de que la idea de este proyecto fue el de realizar un sistema que fuera fácil de instalar, operar y mantener, se puede mencionar que no importa el tipo de sistema que se tenga, lo más difícil es la configuración de los servicios. Puedo mencionar con responsabilidad, que en cierta forma, las personas que instalen el SADNS como tal, mínimo requieren tener conocimientos en Unix, para poder editar y reasignar valores a los archivos de configuración que vienen en el sistema.

La ventaja de este sistema es que el administrador principal instala una sola vez y determina a los administradores de otras redes (usuarios del sistema) que se le otorgan permisos para administrar remotamente sus dominios. Así es que los administradores remotos, pueden accesar a través de Internet utilizando cualquier navegador de Web (que a veces cuando uno accesa a servidores de Web, intuitivamente se conoce el manejo), sin tener grandes conocimientos en Unix y redes. El único requisito que se pudiera mencionar es que los administradores de las redes, conocieran la utilización de los sistemas Windows (de ventanas con mouse).

4.5 PRESENTACIÓN

A continuación se presentan los módulos de acceso que se tienen en el SADNS y la explicación de cada uno de ellos.

El sistema se divide en dos módulos generales, el primero es para cuando accesa el administrador local o principal (admin-local) y el segundo es cuando accesa el administrador remoto o de red (admin-remoto). El primero tiene el control de todo el sistema sobre todo con respecto a la administración del DNS, el segundo controla y administra el (los) dominio(s) que administra remotamente.

El sistema está formado por los módulos de administración. La interface es amigable y fácil de utilizar (basado en el cliente del servidor de Web). La utilización del sistema es un poco intuitivo por la utilización de las ligas de hipertexto del sistema.

En general, puedo mencionar que el Sistema de Automatización del DNS (SADNS) está formado y constituido de la estructura siguiente:

- Módulos de Administración
  - Administrador Local o Administrador Principal (todos los dominios y privilegios sobre el DNS local)
  - Administrador Remoto o de red de uno o más dominios
- Entrada general de información para ambos módulos
- Cada administrador dependiendo de los permisos podrá accesar a diferentes áreas (módulos)
- Ayuda en línea para cualquier módulo del sistema

La estructura del sistema presentados en los puntos anteriores, definen y muestran la flexibilidad de uso. Enseguida vamos a mostrar como se pueden ver los accesos y manejo de cada página en el Web (usando el SADNS con el CGI respectivo).

MÓDULO ADMINISTRADOR LOCAL (ADMIN LOCAL)

Para accesar al sistema del módulo principal, lo que el administrador deber realizar es accesar al URL especificado, es decir, la dirección configurada para accesar al SADNS, esto es que el URL del sistema es *http://www.sadns.mx* . Se puede observar que http, es un URL definido, que implica que se hará una conexión a un servidor de Web. Otros URL definidos son: telnet, ftp, gopher, news y otros que se pueden definir dentro de nuestras aplicaciones.

Para que el Administrador Local o Principal pueda accesar al SADNS se requiere:

1. Acceso a Internet o RedUNAM
2. Conexión desde un cliente Web (Netscape) a *http://www.sadns.mx*
3. Tener la clave y contraseña del Administrador Local.

Se presentan las páginas que el Administrador Local tiene acceso, así como los eventos que obtendrá cuando trate de accesar al sistema.

Teniendo el acceso a un cliente de Web, se abre una conexión a un URL, el definido para nuestro sistema. Como se ve en la figura 4-3.

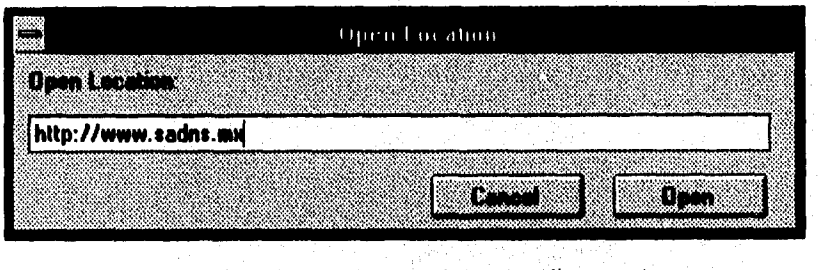


Figura 4-3. Abriendo una Conexión al URL: http://www.sadns.mx

Al invocarse el URL, lo que ocurre es que a través de Internet, el cliente de Web se conecta con el servidor solicitado. Lo que el protocolo de httpd presenta es que como se encuentra configurado con autorización de acceso, aparece una solicitud de clave de acceso y contraseña. Internamente, los CGIs internos para cada módulo valida el acceso del tipo de administrador. Para este caso, se trata del administrador principal, la clave es: *root* y el password es: *sadns03* La caja de diálogo se puede observar en la figura 4-4.

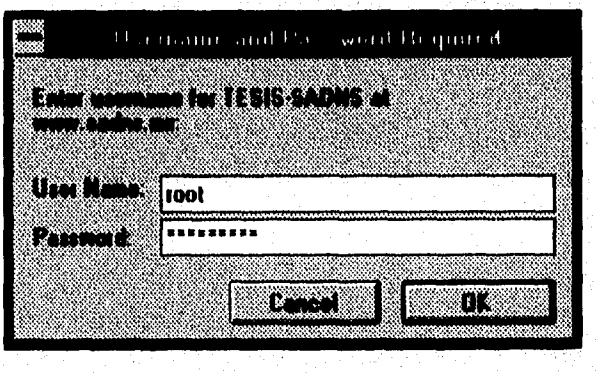


Figura 4-4. Validación del tipo de usuario.

El protocolo valida la entrada del usuario y permite que éste accese a la página inicial general. Se presentan mensajes comunes para cualquiera de los modos de acceso, se presentan escudos de la UNAM y de la Facultad de Ingeniería y un botón de acceso para entrar a los comandos de administración.Figura 4-5.

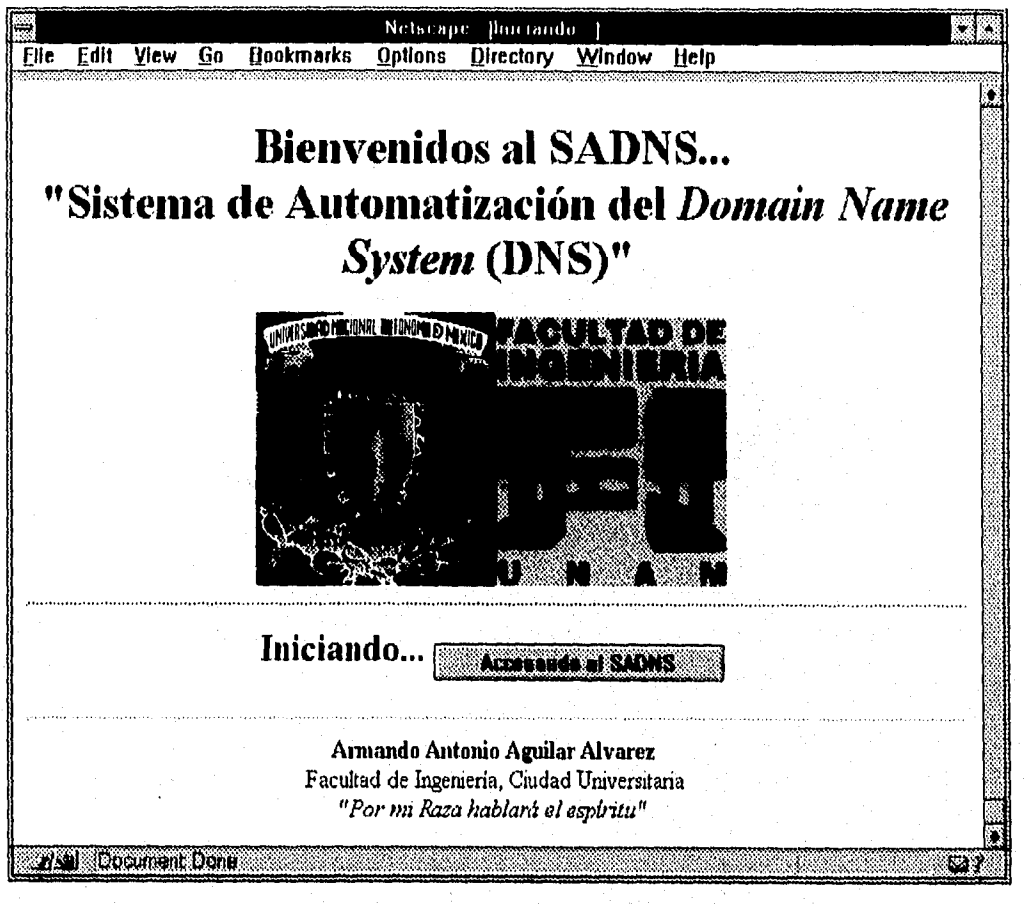


Figura 4-5. Página Inicial y Principal del Sistema.

En la figura 4-6, se pueden observar los comandos correspondientes a los que tiene acceso el usuario en modo de Administrador Local o Principal. Los comandos son:

- Mantenimiento de Dominios
  Este comando permite agregar, borrar, modificar o consultar un dominio tanto por nombre como por dirección inversa dentro del servidor de nombres.

- Mantenimiento de Usuarios
  El administrador principal o local tiene la posibilidad de agregar, eliminar, cambiar o consultar a usuarios (administradores remotos o locales) al SADNS.

- Mantenimiento de Bases de Datos
  Al elegir esta opción, el administrador principal tiene la posibilidad de agregar, borrar, o modificar los tipos de registro del DNS para las bases de datos que se representan por todos los dominios dentro del servidor.

- Activación de Proceso (NAMED)
  En esta opción se realiza el arranque o reinicialización del servidor *named* del DNS, se realiza esta opción para que el servidor de nombres reconozca y tome los cambios realizados en el mantenimiento de las bases de datos de cada dominio administrado. Esta opción la realiza únicamente el usuario local o administrador principal del sistema.

- Respaldos
  El sistema tiene la capacidad de respaldar las bases de datos de cada dominio, así como la configuración del servidor de nombres. Se puede guardar en un archivo compactado o bien, en una cinta de respaldo previamente conectado a la máquina.

- Herramientas del DNS
  Proporciona al administrador preguntas y respuestas sobre el DNS, también permite buscar información sobre la existencia de nombres de máquinas o direcciones dentro de Internet.

- Usando la Ayuda
  El SADNS tiene servicio de ayuda en línea usando páginas HTML (con hipertexto). La ayuda en línea permite elegir la explicación básica y concreta de la pantalla en uso. La ayuda trata de describir los campos que conforman las formas de captura del sistema (interface).

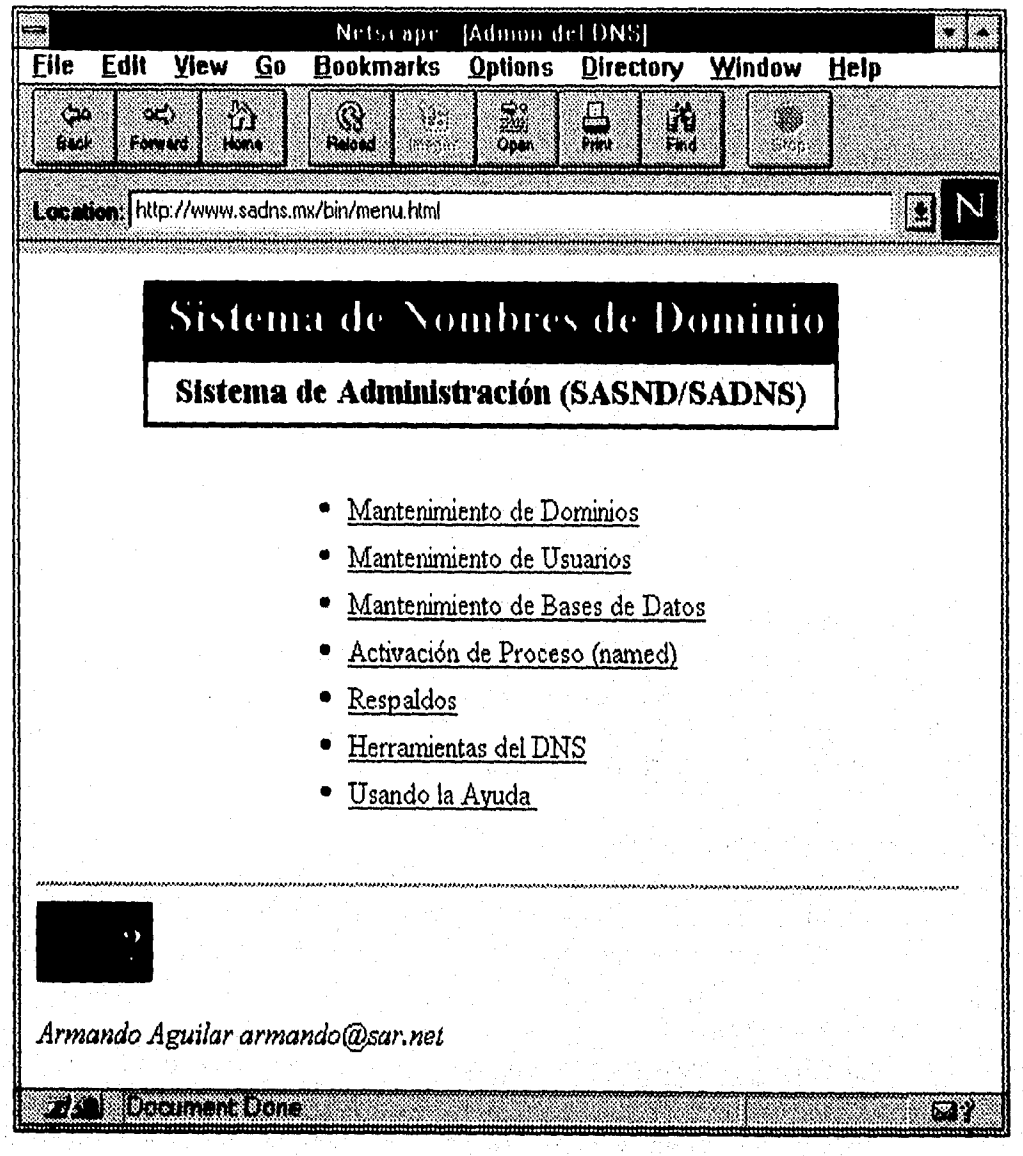


Figura 4-6. Comandos del Administrador Local

En cada opción del módulo del administrador (antes visto), se presenta en la figura 4-7 el comando de Mantenimiento de dominios.

A continuación se definen las opciones que contiene este comando del SADNS.

- Agregar
  Se refiere a que se puede dar de alta a la configuración del servidor de nombres al archivo */etc/named.boot* un servidor primario o secundario por nombre o inverso (dirección).

- Borrar
  La opción permite elegir los dominios (primarios y secundarios por nombre e inverso) que se desean dar de baja del servidor de nombres. Al ser borrados, el administrador remoto se limita a accesar a los dominos que únicamente le queden asignados.

- Cambiar
  Dentro del SADNS existe la posibilidad de realizar los cambios necesarios a un registro existente (creado anteriormente).

- Consultar
  Se pueden consultar los dominios que existen dentro del servidor de nombres configurado. En este mismo contexto, se pueden desplegar los hosts de cada uno de los dominios elegidos.

- Regresar al menú principal
  Esta opción proporciona regresar a la página principal del sistema.

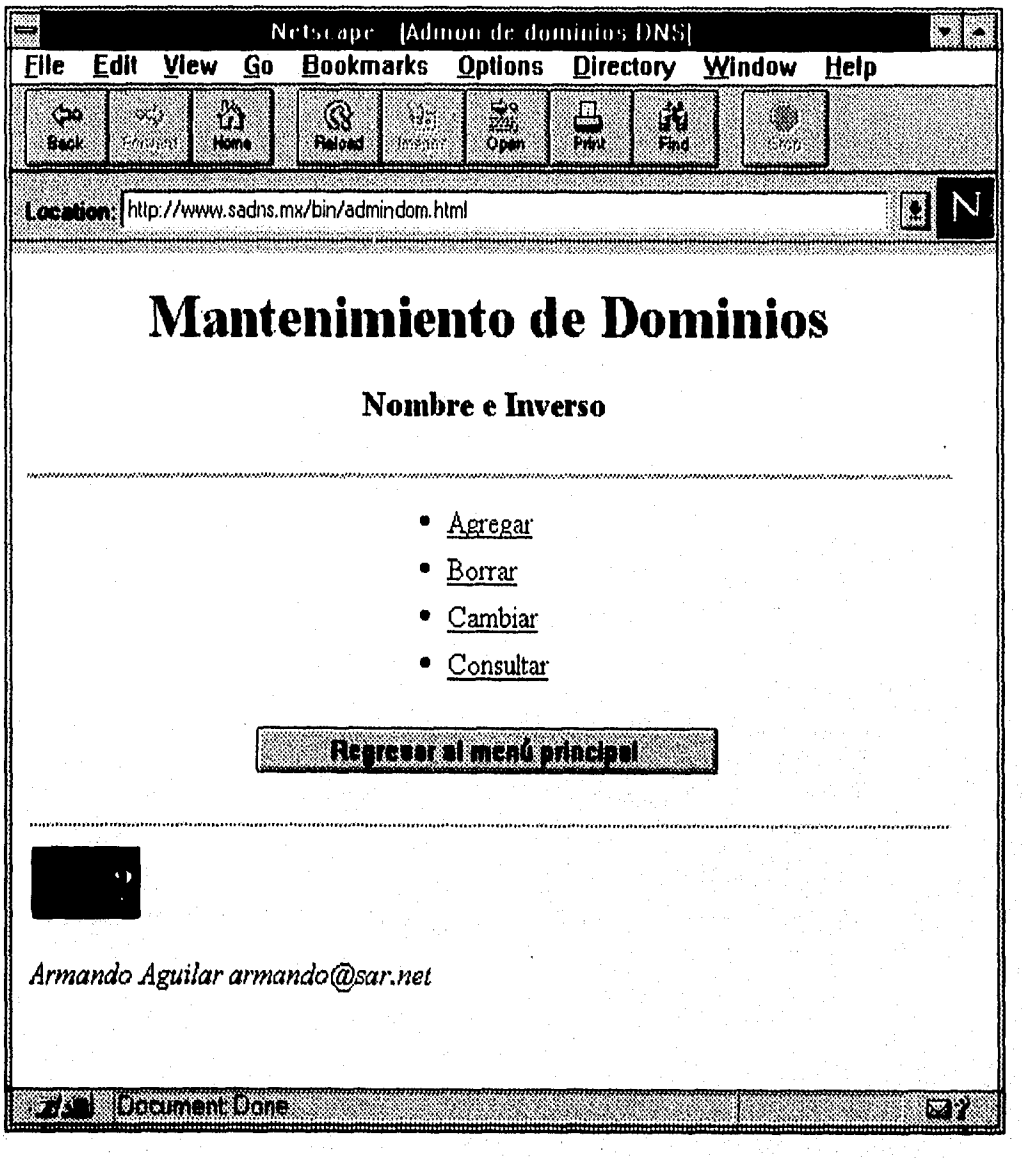


Figura 4-7. Mantenimiento de Dominios.

En la figura 4-8, se presenta el menú de Mantenimiento de Usuarios del SADNS. Se muestran los comandos: Alta, Baja, Actualización y Consultas.

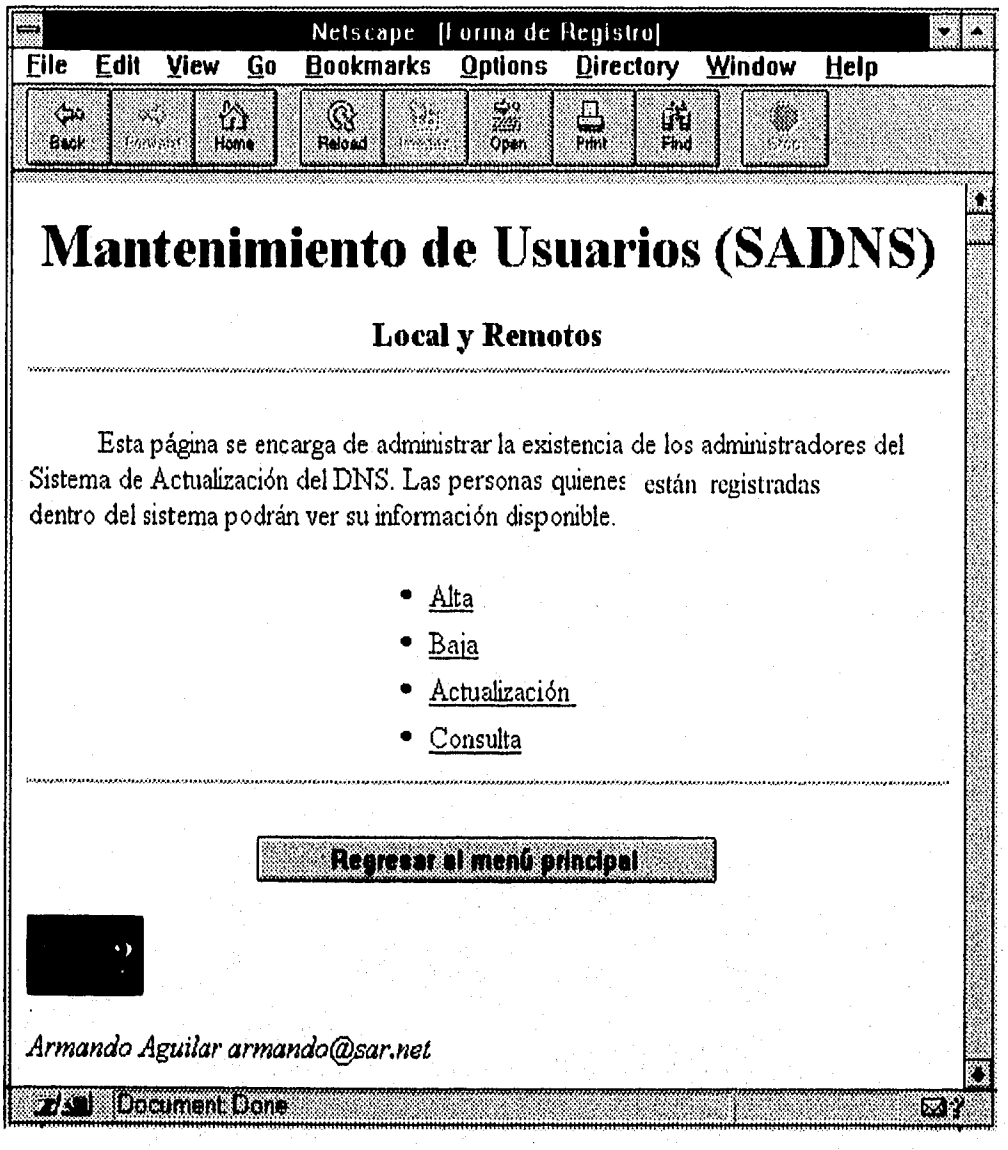


Figura 4-8. Mantenimiento de Usuarios (SADNS)

En este menú de comandos, es importante mencionar que existe validación para cada tarea que se haga por usuario. Los usuarios a dar de alta, se validan con respecto a los que ya existen. La baja de usuarios se valida con respecto a que si el usuario no administra ningún dominio actualmente, entonces se procede a borrar de lo contrario, es necesario que se cambie el administrador del dominio o bien, se borre el dominio. Para el caso de modificación de usuario, ocurre la misma validación, es decir, es como si fuera alta y baja a la vez.

En la figura 4-9 se presenta el comando de Mantenimiento de Bases de Datos. En esta pantalla se despliega el título de Mantenimiento de Dominios por root (donde realmente se refiere a la administración de los archivos de configuración que representa cada dominio dentro de nuestro servidores de nombres. El administrador que está accesando en esta pantalla es el Administrador Local con la clave de *root*

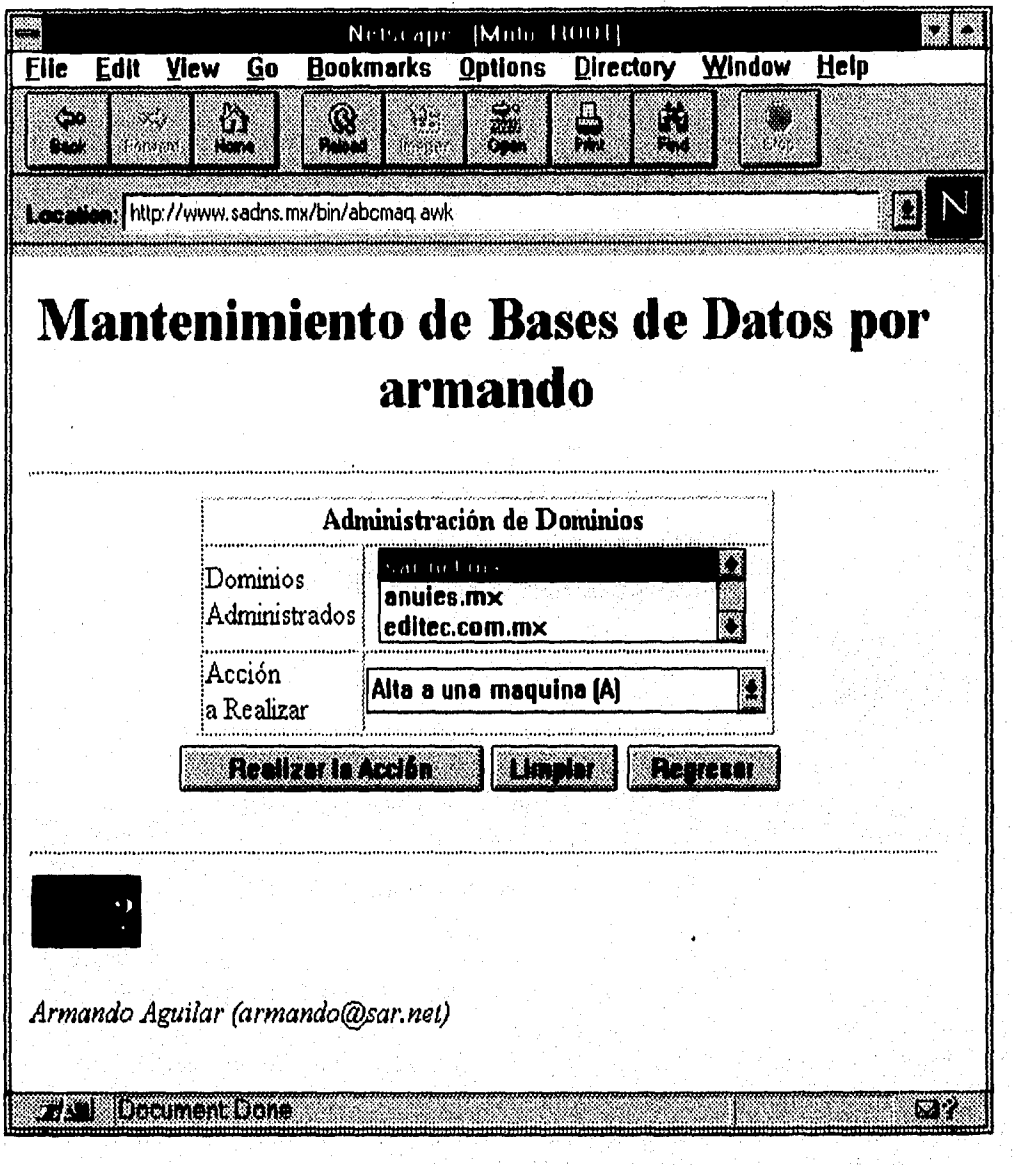


Figura 4-9. Mantenimiento de Bases de Datos

En los títulos de las pantallas que dicen *Dominios Administrados* se refieren a los dominios que el administrador tiene acceso para agregar, eliminar, modificar y consultar elementos de registros. En la figura 4-9 se presenta el Mantenimiento de Bases de Datos con la posibilidad de actualizar tipos de registros del DNS (RR), esta opción se puede utilizar visualizando el título de *Acción a Realizar* donde se despliegan las opciones de dar Altas, Bajas, Cambios o Consultas de algún tipo de registro en particular.

Se muestra en la pantalla el ícono de Ayuda, lo cual uno puede elegirlo para ver la forma de como utilizar el Mantenimiento de Bases de Datos.

En la figura 4-10 se despliega la opción de Activación de Proceso (NAMED). En la pantalla aparece *Activación del Servidor del DNS* (proceso *named*). Como se ha mencionado esta inicialización o reactivación del proceso es con la finalidad de que en el servidor de nombres para su resolución reflejen los cambios realizados o bien puedan obtenerse transferencia de zonas de otros dominios. Se puede observar que este comando tiene las posibilidades de activarse por horario usando (en ambientes Unix el comando *crontab* que realiza la posibilidad de hacer tareas en tiempos distintos. Se aconseja ver el manual para llevar a cabo esto.

También la posibilidad de Activación Inmediata que tiene la posibilidad de reinicializar el servidor usando el proceso actual o bien, iniciarlo con un nuevo proceso en el sistema lo cual significa un diferente comando para cada sistema operativo. Esto es, que si se tiene que activar el proceso del servidor en forma automática por el administrador con solo elegir la opción.

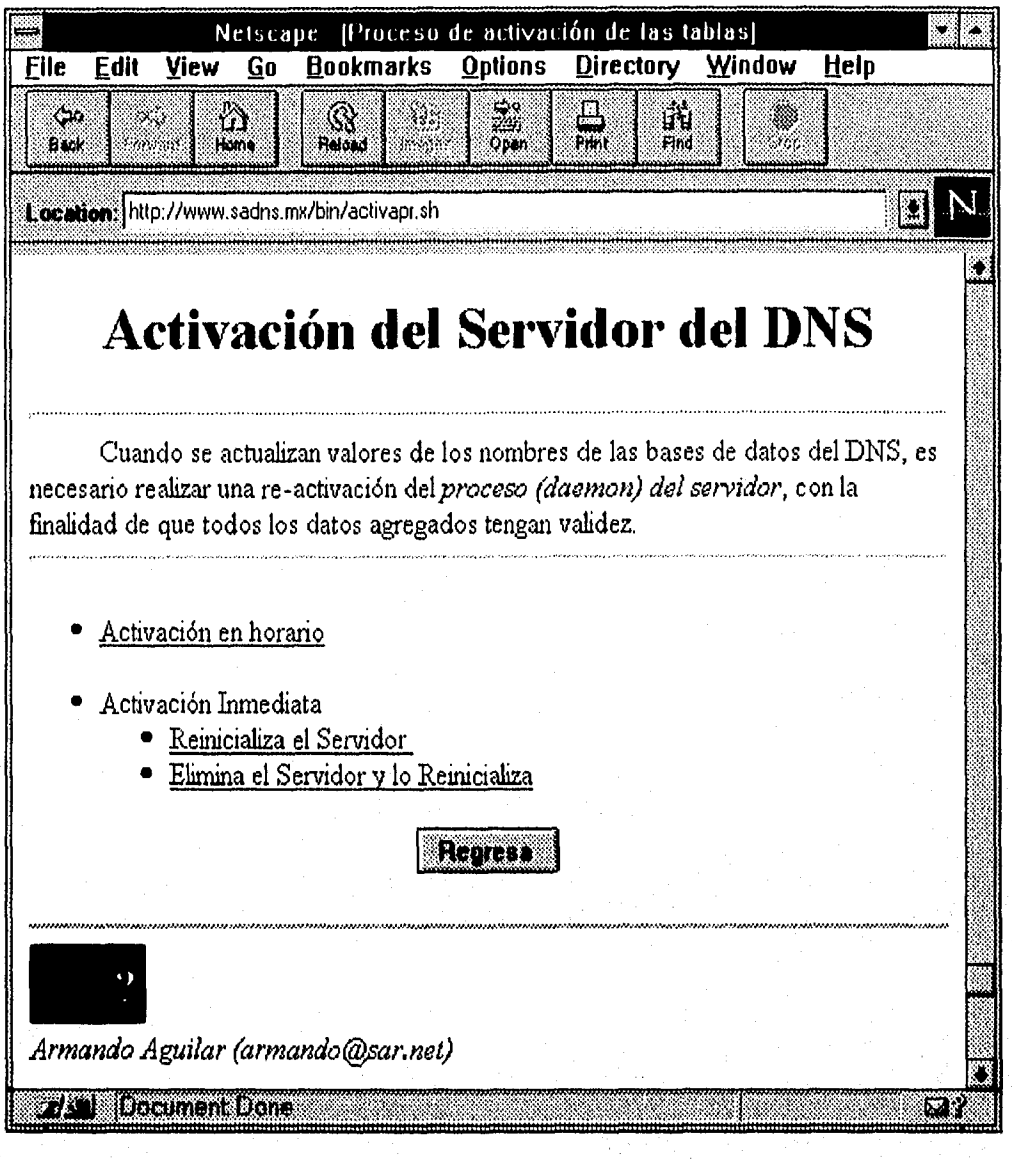


Figura 4-10. Activación de Proceso NAMED

En la figura 4-11 mostramos la pantalla de Herramientas del DNS que no son más que utilerías y documentos que nos proporcionan mayor entendimiento de lo que estamos realizando. También se muestran tips, restricciones y comentarios a propósito del DNS.

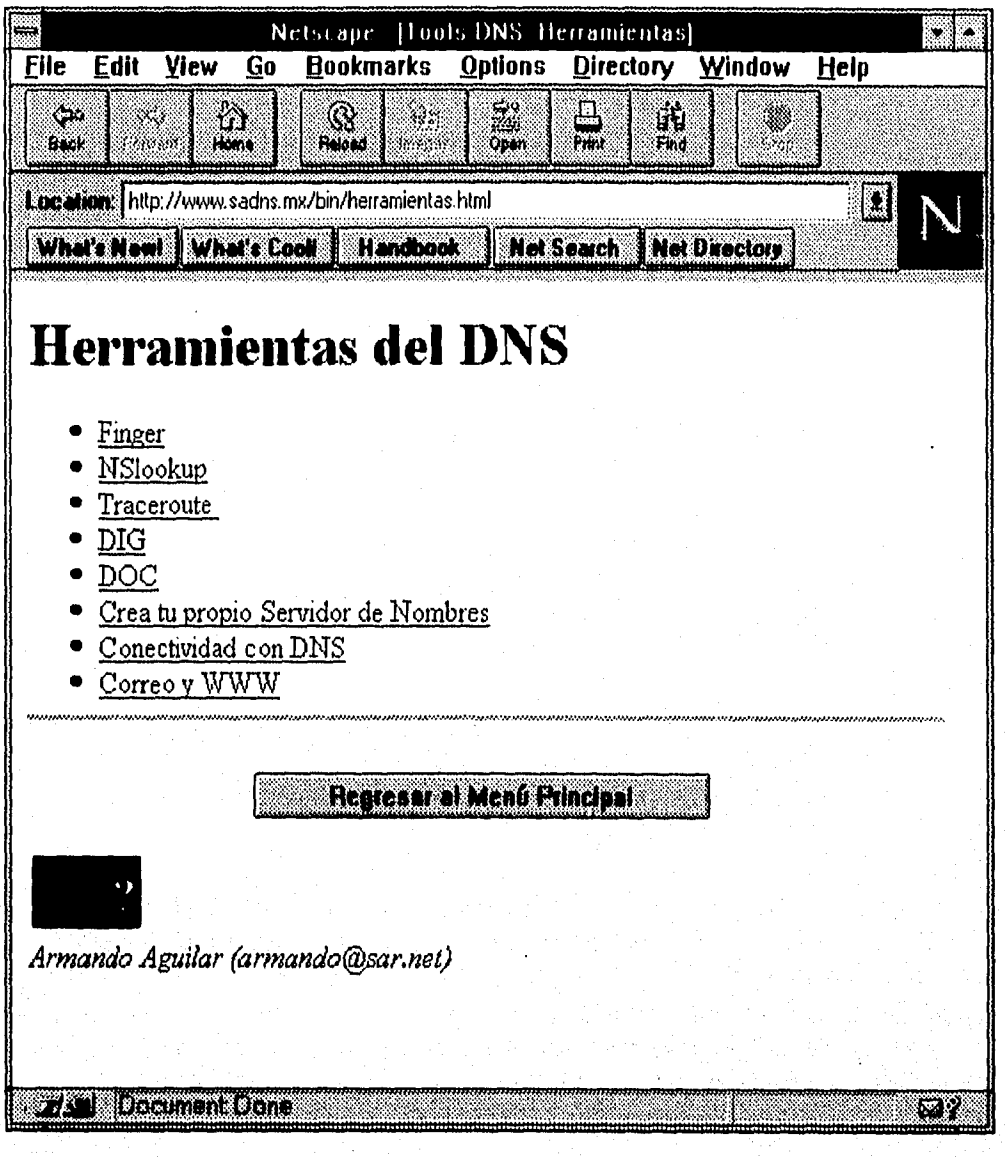


Figura 4-11. Herramientas del DNS

En Herramientas del DNS se presentan las siguientes opciones: Finger, Nslookup, Traceroute, DIG, DOC, Crea tu propio Servidor de Nombres, Conectividad con DNS, Correo y WWW. En cada una de estas opciones se presenta la posibilidad de revisar la ayuda en línea, se presenta una definición y forma de utilizarse.

- Finger es un comando que sirve para ver a usuarios a través de la red. Con este comando podemos verificar la existencia de usuarios en algún

servidor de la red y comprobar que nuestro servidor de nombres tiene la capacidad de resolver nombres de forma aceptable (eficiente y eficazmente).

- Nslookup es una herramienta clave para el buen desempeño del DNS, sirve para realizar verificaciones de las actualizaciones en nuestro servidor de nombres.

- Traceroute es un comando que sirve para ver la trayectoria de nuestra máquina fuente a un destino determinado.

- Crea tu propio Servidor de Nombres es un comando que persigue el fin de dar a conocer a los administradores la posibilidad de crear su propio servidor de nombres dentro de su propia red.

- Conectividad con DNS. Es un documento que expresa la importancia de utilizar DNS dentro de Internet, así como las responsabilidades de los administradores de una red cuando tienen a su cargo uno o más dominios (o bien un servidor primario).

- Correo y WWW. Es un documento que explica la importancia de utilizar el DNS dentro del correo electrónico de Internet y las posibilidades y ventajas de usar el servicio para servidores de WWW.

En la figura 4-12 se muestra la pantalla de realización de Respaldos del SADNS. En esta página el Administrador Local tiene la autoridad de obtener un respaldo de todo el sistema con sus archivos de bases de datos de todos los administradores remotos. Existe la opción de guardar la información en un archivo compactado y comprimido y almacenarse en otro servidor Unix, o bien, guardarse en un dispositivo de respaldo como un *DAT* (unidades de cinta de 4mm).

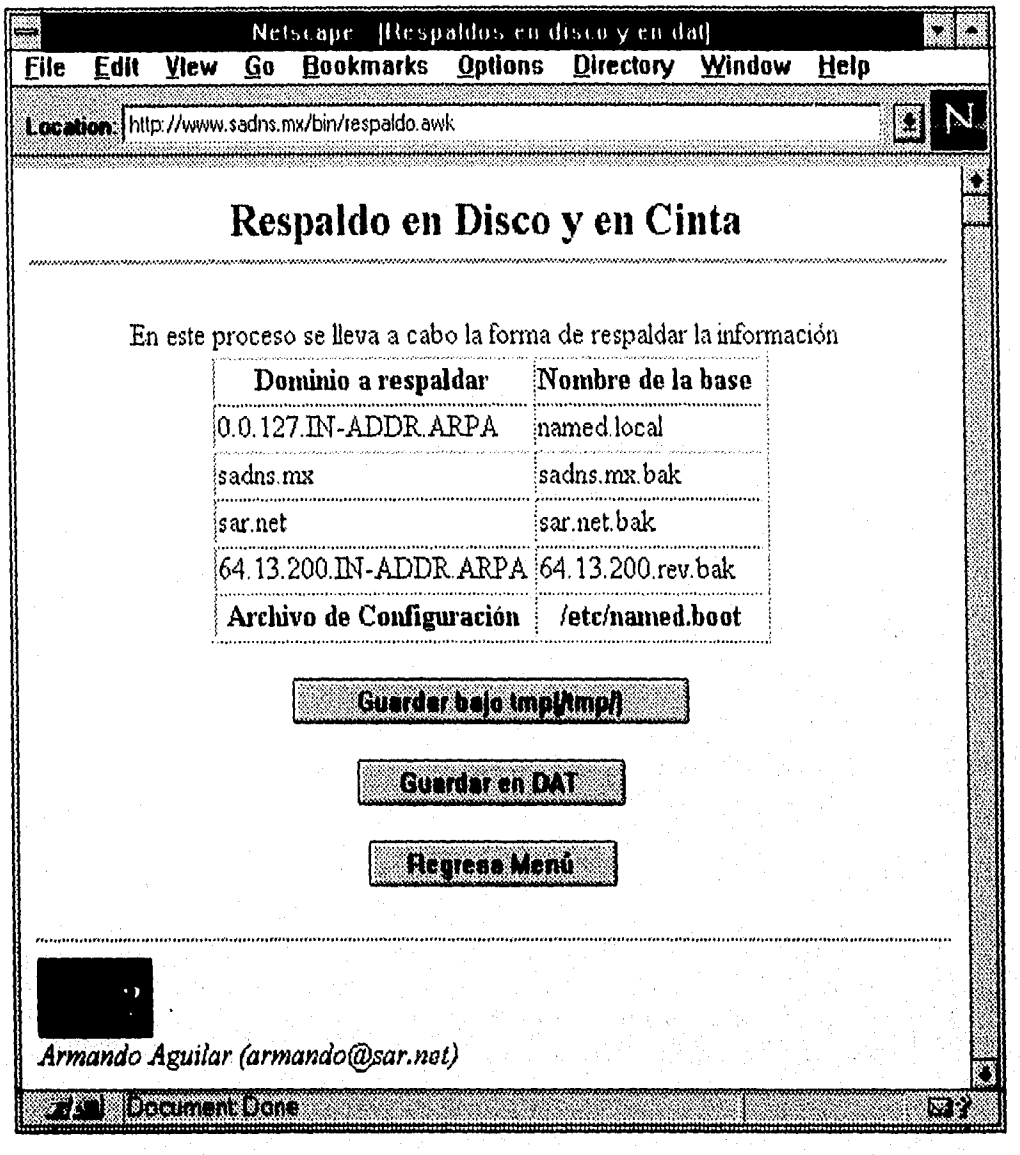


Figura 4-12. Respaldos

Por último en la figura 4-13 se tiene el comando de Usando la Ayuda. En esta página se muestra la forma de utilización del modo de ayuda en línea del SADNS.

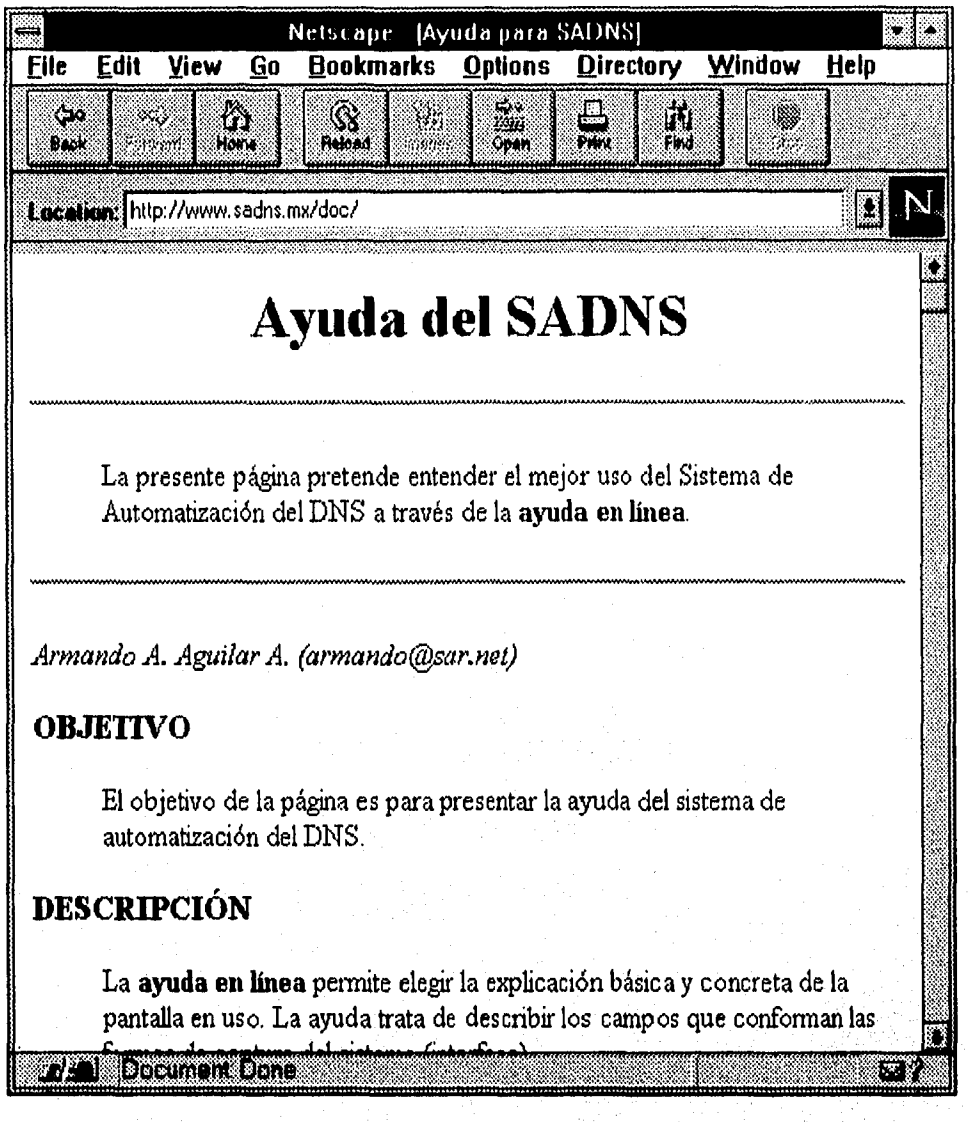


Figura 4-13. Usando la Ayuda

MÓDULO ADMINISTRADOR REMOTO (ADMIN REMOTO)

El módulo Administrador Remoto es similar al módulo de administración del usuario local, pero, existen comentarios interesantes por describir.

Para accesar este módulo es necesario que el administrador remoto haya solicitado previamente su clave de acceso al administrador principal y además de pedir el o los dominios para ser manejados. Esto se resuelve cuando el Administrador Remoto solicita al Administrador Local vía telefónica o

personalmente una clave de acceso al sistema para poder manejar los archivos de bases de datos del SADNS (que son los archivos generados por cada dominio delegado).

Para accesar al SADNS, el Administrador Remoto debe tener lo siguientes:

1. Acceso a Internet o REDUNAM
2. Conexión desde un cliente Web a *http://www.sadns.mx*
3. Tener la clave y constraseña del Administrador Remoto (otorgada por el Administrador local)

Se presenta a continuación la secuencia de páginas accesadas por cualquier Adminitrador Remoto al sistema. Como se ha mencionado en repetidas ocasiones, este módulo es accesado por varios Administradores Remotos que tienen la capacidad de realizar el mantenimiento de sus bases de datos delegadas por el Administrador Local o Principal.

Desde un cliente o navegador del WWW (Netscape®) conectarse al URL definido para nuestro sistema: *http://www.sadns.mx*. Ver la figura 4-15.

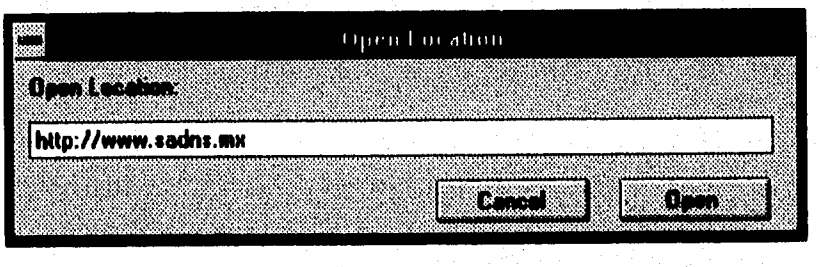


Figura 4-14. Entrando al URL del SADNS

Al invocarse la conexión aparece una ventana donde solicita la clave del tipo de administrador que entrará al SADNS. Ver figura 4-15.

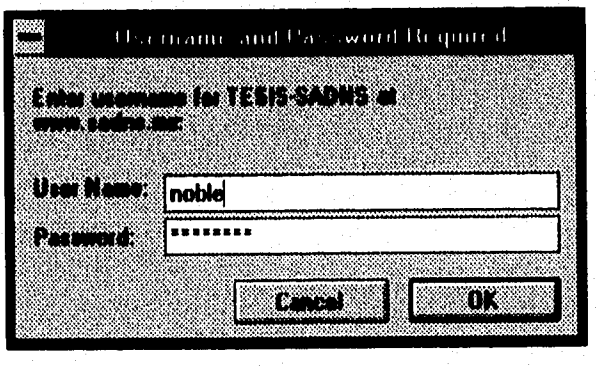


Figura 4-15. Validación del Tipo de Usuario (Admin Remoto)

Al validar el acceso, el sistema permite entrar al usuario remoto a la pantalla mostrada en la figura 4-5 presentada en este capítulo y después de oprimir con el ratón el botón Accesando al SADNS aparecerán los principales comandos que puede manejar cualquier Administrador Remoto. Como se observa en la figura 4-16. Los comando son:

- Mantenimiento de Bases de Datos
  En este comando el Administrador Remoto tiene la posibilidad de mantener a uno o más dominios según le sean delegados o bien autorizados. En esta parte del sistema se realizan las altas, bajas, cambios o consultas de los tipos de registro del DNS. Cuando un administrador elige esta opción, el sistema automáticamente desplegará en la pantalla los dominios que el Administrador Remoto puede manejar.

- Aviso al Administrador Local
  Al ejecutarse esta opción el Administrador Remoto tiene la posiblidad de comunicar a través de correo electrónico al Administrador Local que realice la activación del servidor de nombres, cabe mencionar que el usuario remoto no tiene esta capacidad, se reserva para el Administrador Local (de acuerdo a su aprobación).

- Respaldos
  El Administrador Remoto tiene la capacidad de realizar respaldos de sus archivos de bases de datos generados por cada dominio administrado guardando cada uno de ellos en su disco local o bien pueden enviarse cada archivo por correo electrónico.

- Herramientas del DNS
  Este comando tiene las mismas características tanto para el Administrador Local como para el Administrador Remoto.

- Usando la Ayuda
  El SADNS tiene el servicio de ayuda en línea para este módulo. Usando hipertexto se relacionan los tópicos para cada página y la ayuda trata de describir todo el contexto de cada módulo.

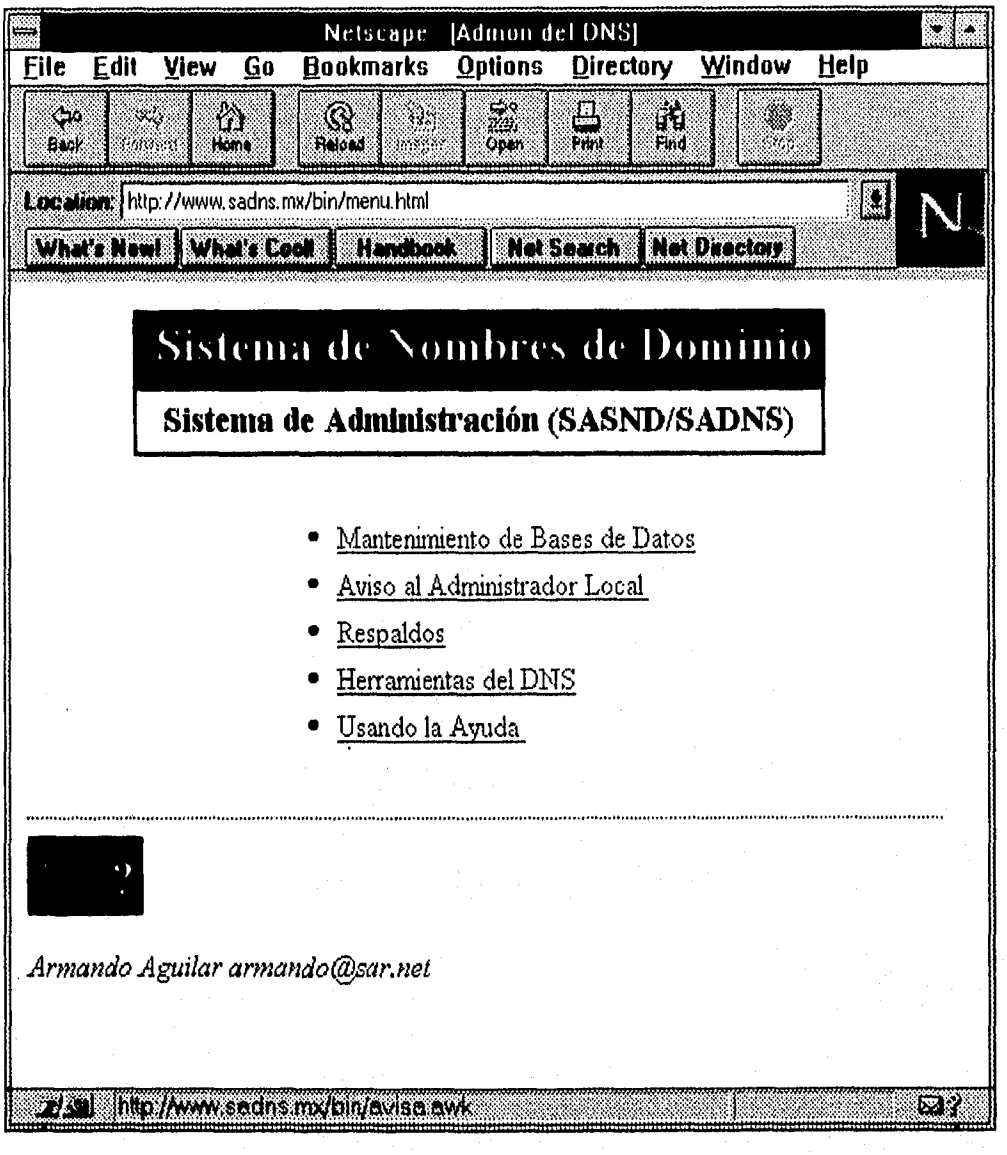


Figura 4-16. Comandos del Administrador Remoto

Se presenta en la figura 4-17 el comando de Mantenimiento de Bases de Datos para este módulo. En esta pantalla se despliegan automáticamente los dominios administrados por cada usuario del sistema. Es importante mencionar que también existe la posibilidad de agregar, borrar, cambiar o consultar cada tipo de registro presentados ahí tales como: A, CNAME, MX, NS, etc.

Al elegir cualquier acción por realizar, se translada el programa a diferentes pantallas que permiten capturar o elegir características para cada dominio.

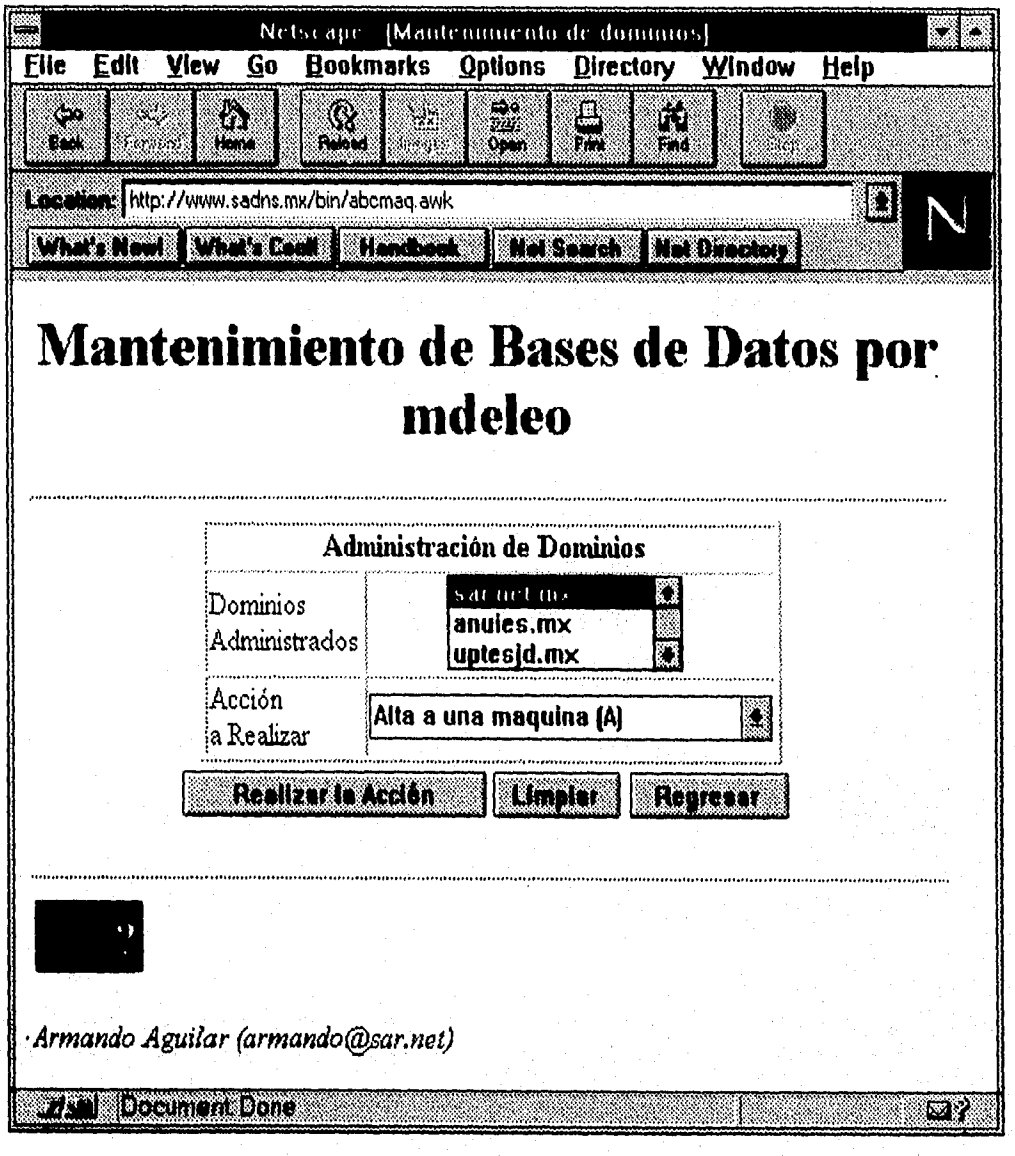


Figura 4-17. Mantenimiento de Bases de Datos

En la figura 4-18 se muestra el comando de Aviso al Administrador Local, se puede observar que existe la posibilidad de escribir los comentarios percibidos por el Administrador Remoto y además se puede mencionar en esa pantalla la necesidad de restablecer el servidor de nombres (*named*) para que el administrador realice la operación. El mensaje se envía por correo electrónico.

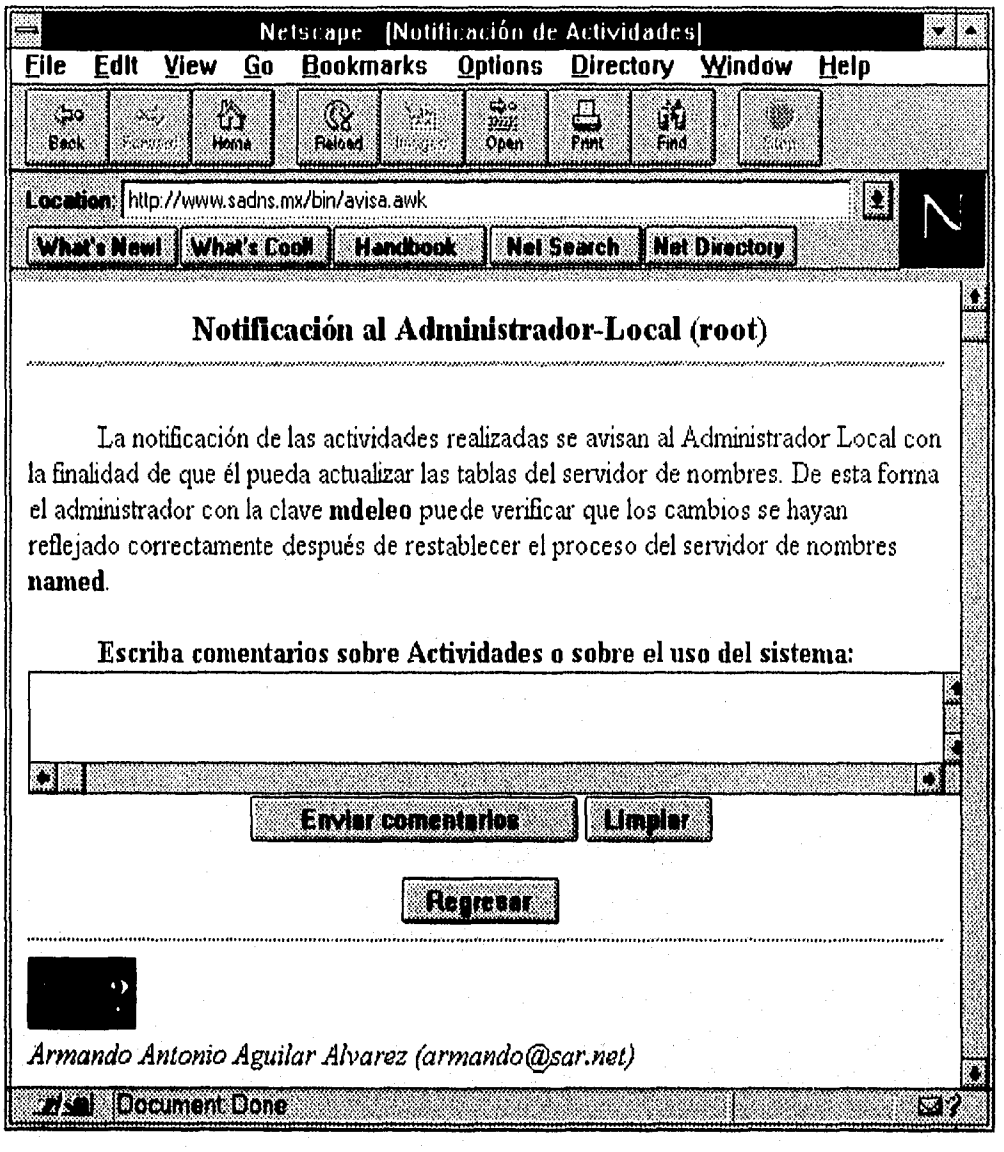


Figura 4-18. Aviso al Administrador Local

En la siguiente figura 4-19, se muestra la opción de Respaldos. En ella se despliega el responsable y la elección de cada dominio para ser guardado en disco o ser enviado por correo electrónico. El respaldo se realiza por responsabilidad propia de cada administrador (aparte que en el módulo del Administrador Remoto se hace de forma general).

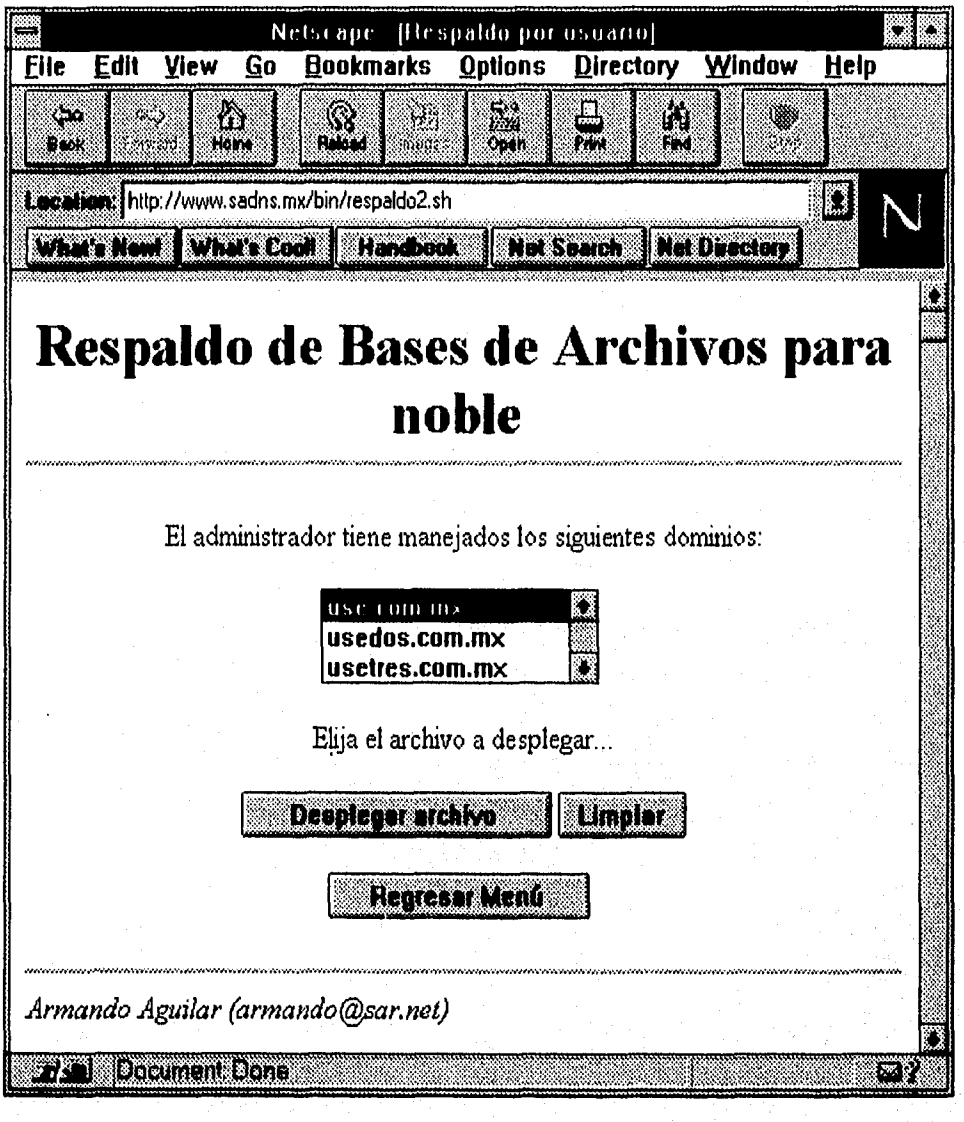


Figura 4-19. Respaldos para el Administrador Remoto

Los comandos de Herrramientas del SADNS y Usando la Ayuda son iguales para los dos módulos Administración Local y Administración Remota. Pueden observarse las figuras 4-11 y 4-13 respectivamente.

En general la posibilidad de administración de un usuario remoto se limita al número de dominios que pueda manejar. Realmente la parte importante de este módulo es el Mantenimiento de las Bases de Datos, que son archivos generados por la creación de dominios en el DNS.

RESUMEN

Se puede resumir que los dos módulos tienen diferentes aspectos en sus variados comandos. Casi podemos mencionar que por diseño el módulo de la Administración Local contiene al módulo de Administración Remota y que esta última es parte de la primera. Es quizá importante recalcar que en esta presentación del SADNS se ha realizado un tanto de forma rápida al presentar los comandos importantes de cada módulo, así como la secuencia de cada uno de ellos, pero, es conveniente mencionar que cada comando (alguno con pantallas de captura) tienen la capacidad de validar los datos con respecto al análisis de cada estructura de datos.

En eventos de carácter de validación o obtención de resultados, el SADNS muestran ciertas pantallas de mensajes de error, con ello controlan las salidas de forma clara y simple. El Administrador Local o Remoto por claridad del mensaje o intuición detalla el tipo de error que se ha cometido, así como algún acierto del mismo.

El Desarrollo del SADNS muestra la conceptualización del sistema de forma global. Este estudio se presenta desde el inicio de la elección del conjunto de protocolos y programas que integran al sistema hasta la finalización y presentación del sistema.

5. Conclusiones

Definitivamente en este proyecto se han obtenido y cubierto resultados muy favorables. Todas la expectativas han sido rebasadas debido al empuje y a la popularidad que Internet a tenido en nuestra década. La facilidad de accesar a sistemas de información y procesar datos han sido de gran utilidad para el ámbito académico y privado.

Debido al uso del Sistema de Automatización del Domain Name System (DNS), se ha visto que es una forma fácil y rápido de producir sistemas de información con grandes volúmenes de datos que pueden ser procesadas por lenguajes y manejadores de bases de datos más sofisticados.

El utilizar el SADNS, es sin duda una gran meta obtenida, debido a que hubo un compromiso de crear y proporcionar un procedimiento útil y sencillo para que los administradores de redes con acceso a Internet (y sin acceso) manipularan su información sobre sus máquinas para el beneficio de sus servicios. Además que es una herramienta que ayuda al administrador a manejar su información sin ser expertos en los servicios de Internet.

El SADNS, es una realidad convertida en hechos que reproducen los objetivos de este trabajo, además que este proyecto invita a desarrollar otras aplicaciones propias de la administración de sistemas a un nivel empresarial o académico.

La difusión de las tecnologías y protocolos, así como Internet que es una red de servicios, requieren de procedimientos de divulgación para la comunidad. Este trabajo se traduce pues, en un documento básico de configuración y simplificación de la administración y mantenimiento del Sistema de Nombres de Dominio.

Este trabajo no pudo cubrir todos los servicios que el DNS tiene, ya que hablar de tecnología es cambio constante. El DNS aún está en desarrollo, es decir que todavía está en evolución y día a día están saliendo nuevas versiones del software. Hoy en día se están investigando sobre nuevas tecnologías para aplicarse a Internet y por consecuencia al DNS. Los desarrollos recientes sobre DNS tienen gran enfoque en base a la seguridad en contra de violaciones a sistemas.

Pienso que este trabajo no es terminal, sino que es importante aprovechar los recursos que se tienen para mejorar e implantar sistemas de administración de servicios.

BIBLIOGRAFÍA

Comer, Douglas E., "Internetworking with TCP/IP". Volumen I.
Principles, Protocols and Architectures Second Edition.
Prentice Hall. USA. 1991.

Albitz, Paul; Liu, Cricket. "DNS and Bind".
O'Reilly & Associates, Inc. USA 1992.

Craig, Hunt. "TCP/IP Network Administration".
O'Reilly & Associates, Inc. USA 1992.

Kernighan, Brian; Ritchie, Dennis. "El lenguaje de programación C".
Segunda Edición.
Prentice Hall. México 1991.

Tenenbaum, Aaron; Augenstein, Mosche. "Estructura de Datos en Pascal".
Prentice Hall. México 1983.

Fairley, Richard. "Ingeniería de Software".
McGraw Hill. México 1988.

Pressman, Roger. "Ingeniería del Software, un enfoque práctico". Tercera edición.
McGraw Hill. México 1993.

McClure, Carma. "CASE la Automatización del Software".
Addison Wesley Iberoamericana. USA 1993.

Gane, Chris; Sarson Trish. "Structured Systems Analysis: Tools and Techniques".
Improved Sytstem Technologies, Inc. USA.

Costales, Brian; Allman Eric; Rickert, Neil. "Sendmail".
O'Reilly and Associates, Inc. USA 1993.

Vixie, Paul; Dunlap, Kevin. "Name Server Operations Guide for BIND". Release 4.9.3.
Internet Software Consortium. La Honda, CA. USA.

Stahl, M. "Domain Administrator Guide". Internet Request for Comment 1032 (RFC 1032).
Network Information Center, SRI International, Menlo Park,CA. November 1987.

Lottor, M. "Domain Administrators Guide". Internet Request for Comment 1033 (RFC 1033).
Network Information Center, SRI International, Menlo Park,CA. November 1987.

Mockapetris, Paul. "Domain Names-Concept and Facilities." Internet Request for Comment 1034 (RFC 1034).
Network Information Center, SRI International, Menlo Park,CA. November 1987.

Mockapetris, Paul. "Domain Names-Implementation and Specification." Internet Request for Comment 1035 (RFC 1035).
Network Information Center, SRI International, Menlo Park,CA. November 1987.

Everhart, C.; Mamakos, L.; Ullmann R.; Mockapetris, P. "New DNS RR Definitions" Internet Request for Comment 1183 (RFC 1183).
Network Information Center, SRI International, Menlo Park,CA. October 1990.

Border Network Technologies, "The BorderWare Firewall Server, User's Guide. Version 3.1."
BorderWare Border Network Technologies, Inc. Canadá Noviembre 1995.

net.Genesis; Hall, Devra. "Build a Web Site. The Programmer's Guide to Creating, Building, an Maintaining a Web Presence."
Prima Publishing, Inc. USA 1995.

Lemay, Laura. "Teach Yourself Web Publishing with HTML in a Week". Primera Edición.
SAMS Publishing, Inc. USA 1995.

Badgett, T.; Sandler, C. "Welcome to... Internet: From Mystery to Mastery".
MIS:Press. USA 1993.

Tolhurst, A. William; Pike, Mary Ann; Blanton, Keith. "Using the Internet".
QUE. USA 1994.

ANEXO A. CONFIGURACIÓN DE SERVIDORES

A.1 DNS Y BIND

El servidor de nombres es un programa que corre sobre un *host* que está siempre en espera de recibir solicitudes y a su vez responder o enviar las peticiones a otros servidores. Para tener un programa que funcione de servidor de nombres, tiene que instalarse o bien, compilarse según sea el caso, en algunas versiones de Unix el servidor de nombres ya viene incluído en las utilerías de TCP/IP, tal es el caso de sistemas de arquitecturas como: SUN, SGI, VMS, DEC, etc. En algunos casos, es necesario comprar los grupos de programas que utilizan TCP/IP.

En este trabajo la parte experimental se ha desarrollado sobre UNIX, lo que implica que se ha utilizado el software BIND (*Berkeley Internet Name Domain*), que es un conjunto de programas y librerías que tienen un sistema de cliente/servidor propiamente implantado para satisfacer las características del DNS. En UNIX, el software BIND es el utilizado para este tipo de aplicaciones.

Al hablar en BIND, en términos técnicos, el cliente se llama *resolvedor* y el servidor es un programa (*daemon*) conocido como *named.*

La versión de software BIND se encuentra en los siguientes sitios de ftp:

- *ftp://ftp.vixie.com/pub/dns/bind.4.3.9.tar.Z*
- *ftp://ftp.uu.net/pub/dns/bind.4.3.9.tar.Z*

Algunas listas de discusión sobre el DNS usando BIND en UNIX y otras plataformas:

- *namedroppers-request@nic.ddn.mil*
- *bind-request@vangogh.cs.berkeley.edu*
- *dns-request@listas.sar.net*

Nota: Estas listas requieren de inscripción a través del correo electrónico.

En este anexo se tiene como objetivo poder dar a conocer las diferentes configuraciones de un *cliente* o *servidor de nombres (primario, secundario o cache)* y la construccción de las bases de datos del sistema basados en el software BIND sobre el sistema operativo UNIX. Es importante mencionar que los ejemplos mostrados en esta sección son tomados del lugar donde se desarrolló el trabajo descrito en esta tesis.

A.2 CONFIGURACIÓN DE UN CLIENTE

Un cliente es aquel que tiene configurado lo necesario para solicitar datos a un servidor determinado.

Para configurar un cliente se necesita crear un archivo especial llamado ***resolv.conf*** que debe estar en un lugar públicamente leible bajo el subdirectorio */etc*. Las rutinas del *cliente o resolvedor* lo primero que leen es este archivo, interpretando las siguientes características:

1. A que dominio pertenece el *host* configurado
- ¿Cuál es el servidor de nombres que responderá a las solicitudes ?
- El orden de resolución de la configuración del cliente (*tabla de hosts, nis o dns*)

En el nuevo archivo creado o editado */etc/resolv.conf* se tienen que considerar algunas comandos de configuración. Los comandos soportados por las arquitecturas de UNIX son:

- **domain** *nombre*

  Esta directiva determina el nombre de dominio de default del *host* en el que se está configurando. Al *host* se le agrega el dominio configurado en esta directiva.

- **nameserver** *direcciónIP*

  Esta directiva está apuntando a una dirección IP del servidor de nombres que responderá las solicitudes del las aplicaciones que se corran en el *host*. En las implantaciones de BIND soporta hasta 3 servidores listados en el archivo resolv.conf. para resolución.

Es importante mencionar otra directiva que determina el buen funcionamiento, es cuando en algunos de los sistemas como SGI tienen soportados la directiva *order*.

- **order** *tipo-de-resolución*

  Sirve para determinar el orden de resolución, si el *host* utiliza tabla de *hosts*, *nis* o bien, *dns*.

Se debe mencionar que en algunas plataformas como SUN es necesario modificar otro archivo para que las librerías funcionen (en algunas versiones de su sistema operativo), para resolver nombres, tal es el caso del archivo ***/etc/nsswitch.conf,*** en este archivo se encuentra la configuración general de lo soportado por *Solaris 2.X*, lo que se tiene que agregar en la línea donde aparece `hosts:` `nis`, es la palabra se tiene que agregar la palabra `dns`.

```
hosts:      dns nis
```

Figura A-1. Archivo *nswitch.conf*

Una forma típica de configuración del archivo ***resolv.conf*** es como esta:

```
;Nota: Si no existe /etc/resolv.conf, se crea.
domain      sar.net
order       hosts dns
nameserver  200.13.64.1
nameserver  200.13.64.2
nameserver  132.248.10.2
```

Figura A-2. Archivo */etc/resolv.conf*

La creación o edición de el archivo ***/etc/resolv.conf*** se puede realizar con el editor favorito de cada usuario.

Cada cliente tiene un tiempo fuera, la interpretación del archivo es en orden secuencial, si el primer servidor de nombres no resuelve o envía un error, la petición se pasa al siguiente servidor, si en este ocurre lo mismo se le pasa al siguiente. Esto es en caso de que existieran 3 servidores listados en el archivo.

En resumen, la creación de un cliente es relativamente fácil, únicamente al crear un archivo ***/etc/resolv.conf*** y escribir las directivas con sus datos respectivos.

A.3 CONFIGURACIÓN DE SERVIDORES

Para realizar la configuración de un servidor se requiere considerar un conjunto de archivos cuyos nombres son genéricos, pero se pueden elegir al gusto del administrador, tomando en cuenta algunas consideraciones de configuración estos son:

- **named.boot**
- **named.ca**
- **named.local**
- **named.hosts**
- **named.rev**

Estos archivos están construídos con tipos de registros y comandos que se interpretan de forma que todos están relacionados para el funcionamiento del servidor.

DEFINICIÓN DE ARCHIVOS

Archivo ***NAMED.BOOT***

El archivo named.boot contiene directivas de inicialización que indican en donde se encuentran las fuentes de información, pudiendo ser estas bases de datos locales o bien, de servidores remotos. Por default se encuentra bajo el subdirectorio ***/etc***.

El *named.boot* contiene ciertos comandos o tipos de registros:

directory *directorio-local*
Se define el directorio local utilizado para colocar los archivos de información que el servidor requiera.

cache *nombre-archivo*
Esta directiva o comando apunta al archivo utilizado para inicializar el servidor cache, en sí en este archivo se tiene la lista de servidores de dominio raíz.

primary *nombre-dominio* *nombre-archivo-bd*
Declara al servidor de nombre local como un servidor maestro primario para el domino especificado *nombre-dominio* y carga los datos de la base de datos del archivo *nombre-archivo-bd*. El nombre-dominio puede ser de *dominio* de *nombre* o *dominio inverso*.

secondary *nombre-dominio* *direcciónIP-ns* *nombre-archivo.bak*

Hace que el servidor local sea un servidor secundario maestro para el dominio especificado por *nombre-dominio, la direccionIP-ns* indica la dirección del servidor primario para el dominio que servirá de secundario. El archivo *nombre-archivo.bak* es donde se almacenará la copia de la información del servidor primario cuando se realice un *transferencia de zona.*

forwarders *direccionIP-ns* ...

En este comando se declara una lista de servidores de nombres que pueden resolver peticiones si el servidor local no puede resolver de su propio cache.

slave

Este comando forza al servidor local a utilizar sólo los servidores que están listados junto con el comando de *forwarders.* Este comando se usará solamente si en el archivo *named.boot* existe el comando *forwarders.* Se utiliza cuando existen limitaciones de acceso en la red y que el servidor local puede accesar únicamente a la lista de *forwarders.*

La combinación de los anteriores comandos puestos en el archivo *named.boot* generan *servidores primarios, secundarios o de solo cache.*

Archivo ***NAMED.CA***

Este archivo contiene las direcciones y los nombres de los servidores de nombres raíz; estos son los servidores de ambos sentidos, primero y último en forma inversa. Cuando cualquier servidor se inicializa, primero contacta uno de estos servidores para determinar la o las direcciones de los servidores para su dominio local o bien para determinar quien tiene la autoridad para proporcionar la información.

La forma como se encuentra escrito este archivo esta en un formato para el DNS, propiamente para el archivo cache. El servidor interpreta cada línea como sigue, en el caso de que se tengan las líneas que a continuación se presentan:

```
;       @(#)root.cache  1.15    (Berkeley)      89/09/18
;
; Initial cache data for root domain servers.
;
.                       99999999        IN      NS      NS.INTERNIC.NET.
                        99999999        IN      NS      NS.NIC.DDN.MIL.
                        99999999        IN      NS      KAVA.NISC.SRI.COM.
                        99999999        IN      NS      NIC.NORDU.NET.
                        99999999        IN      NS      NS.NASA.GOV.
                        99999999        IN      NS      TERP.UMD.EDU.
                        99999999        IN      NS      A.ISI.EDU.
                        99999999        IN      NS      AOS.BRL.MIL.
                        99999999        IN      NS      GUNTER-
ADAM.AF.MIL.
                        99999999        IN      NS      C.NYSER.NET.
;
;  Prep the cache (hotwire the addresses).  Order does not matter
;
NS.INTERNIC.NET.        99999999        IN      A       198.41.0.4
NS.NIC.DDN.MIL.         99999999        IN      A       192.112.36.4
KAVA.NISC.SRI.COM.      99999999        IN      A       192.33.33.24
NIC.NORDU.NET.          99999999        IN      A       192.36.148.17
NS.NASA.GOV.            99999999        IN      A       128.102.16.10
A.ISI.EDU.              99999999        IN      A       26.3.0.103
AOS.BRL.MIL.            99999999        IN      A       128.20.1.2
AOS.BRL.MIL.            99999999        IN      A       192.5.25.82
GUNTER-ADAM.AF.MIL.     99999999        IN      A       26.1.0.13
C.NYSER.NET.            99999999        IN      A       192.33.4.12
TERP.UMD.EDU.           99999999        IN      A       128.8.10.90
```

Figura A-3. Archivo *named.ca* (Archivo de Servidores Raíz)

El nombre de dominio "." se refiere al *dominio raíz (dominio root).* Este archivo no necesita actualización, antes, se actualizaba mediante peticiones a foros de interés y se reenvíaban por correo. Como vemos este archivo está formado por una lista de servidores de nombres raíz , es decir, servidores que contienen la información de todos los dominios que se crean y por lo tanto representan y tienen un lugar importante en la configuración global del DNS, por lo que si uno modifica este archivo colocando servidores locales, puede funcionar, pero se recomienda que no se haga dicho cambio.

Volviendo al punto de la actualización, este archivo tiene un número 99999999 que es muy grande y se usa como el tiempo en el que un servidor puede contener registros de información, por eso, el ttl se puede descartar.

En este archivo, se puede ver que esta formado por dominios raíz que apuntan a nombres, y nombres que apuntan a direcciones, dentro del tipo de red IN (Internet).

Una lista actualizada de los servidores raíz estará disponible en algunos lugares como en el Centro de Información de la Red (NIC-INTERNIC) vía ftp en:

ftp://rs.internic.net/pub/domain/named.root
ftp://ftp.sar.net/dnstnt/named.ca.nuevo

Archivo **NAMED.LOCAL**

El archivo *named.local* es un archivo donde se representa la dirección especial que usan los *hosts*, para el tráfico directo a ellos mismos o tan bien conocida como la dirección de *loopback*. Este archivo es importante para establecer la existencia del *host* que quiere ser un servidor de cualquier tipo. El archivo *named.local* representa la dirección inversa con un tipo de registro *ptr* que apunta a el nombre local del *host* en que se está corriendo esta configuración. A continuación se ve la configuración de este archivo y que debe estar en algún determinado dentro del archivo de configuración ***/etc/named.boot***.

```
@ in soa dns.sar.net.  armando.sar.net. (
                         9501.2001  ; yymm.dd[0-9] [0-9]serial
                         3600       ; refresh
                         3600       ; retry
                         604800     ; expire
                         172800 )   ; minimum

                        in    ns    dns.sar.net.
                        in    ns    ns.mci.net.

1     in    ptr   localhost.
```

Figura A-4. Archivo *named.local*

Se puede observar que la línea en el siguiente ejemplo, indica el dominio inverso de `1.0.0.127.in-addr.arpa` declarado en el archivo */etc/named.boot* y su vez se determina que la la dirección `127.0.0.1` tiene el nombre localhost.

```
1     in    ptr   localhost.
```

Figura A-5. Ejemplo de Dominio Inverso Local

El archivo named.local siempre estará formado de esa forma, lo único que cambiará es la configuración del registro SOA, ver el la sección 1.3.6.

Archivo **NAMED.HOSTS**

Propiamente el archivo *named.hosts* es la base de datos del servidor primario maestro, que contiene información del dominio. Este archivo convierte los nombres de los hosts a direcciones IP, así que los tipos de registros que predominan son los tipo A, pero también contiene registros tipo MX, CNAME, HINFO, etc. Es importante recalcar que este archivo solo es creado en el servidor primario.

A continuación se muestra un archivo de *named.hosts* básico y pequeño que nos permita ejemplificar y obtener la estructura de un archivo *named.hosts*.

```
@     in    soa   dns.sar.net       armando.sar.net   (
                        9603.0101   ; yymm.dd[0-9] [0-9]serial
                        3600        ; refresh
                        3600        ; retry
                        604800      ; expire
                        172800   )  ; minimum

                        in    ns    dns.sar.net.
                        in    ns    ns.mci.net.

;El localhost con dirección IP 127.0.0.1 no es necesario ponerlo debido
; a que es suficiente con el archivo named.local
$origin     sar.net.
sar.net.    in    mx    10          chajul.sar.net.
www.sar.net.      in    cname       chajul.sar.net.
dns         in    a     200.13.64.1
chajul      in    a     200.13.64.1
            in    hinfo       SGI irix5.3
solar       in    a     200.13.64.2
gw-solar    in    a     200.13.64.254
slip1       in    a     200.13.68.1
slip2       in    a     200.13.68.2
slip3       in    a     200.13.68.3
cs          in    a     200.13.68.254

$origin     cotz.sar.net.
cotz.sar.net.     in    mx    10    correo.cotz.sar.net.
correo            in    a     200.13.76.1
quija             in    a     200.13.76.25
gw-cotz           in    a     200.13.76.254
```

Figura A-6. Archivo *named.hosts* del Dominio *sar.net*

Se puede observar que en el archivo en las declaraciones de los tipos A, existen más de una clase C implicada en el nombre del dominio, estas son: `200.13.64.0`, `200.13.68.0` y `200.13.76.0`, por lo que el archivo de *named.hosts* puede mantener sobre un dominio, varios tipos de redes.

Archivo ***NAMED.REV***

El archivo *named.rev* tiene la estructura similar al *named.local*, estos archivos hacen la traducción de direcciones IP a nombres de *hosts*, es decir, se tiene el *dominio inverso* ambos archivos contienen registros de tipo PTR. En estos archivos los tipos de registros que intervienene son SOA, NS y PTR.

En estos archivos se define la dirección IP apuntando a un nombre completo del host. Un solo nombre completo es válido para una dirección IP.

Las bases de datos de dominio inverso dependen del número de direcciones de red delegadas para el usuario, es decir, que entre el archivo *named.hosts* y *named.rev* existe la diferencia de que el archivo de nombres puede ser uno solo donde guarde todo el dominio, pero el archivo de dominio inverso puede ser uno para cada tipo de red delegada por el proveedor de Internet. A continuación se presenta un archivo named.rev para una sola dirección de red, el dominio que se va a representar es el `64.13.200.in-addr.arpa`:

```
@      in     soa    dns.sar.net armando.sar.net   (
                          9603.0101    ; yymm.dd[0-9] [0-9]serial
                          3600         ; refresh
                          3600         ; retry
                          604800       ; expire
                          172800    )  ; minimum

                          in     ns     dns.sar.net.
                          in     ns.    ns.mci.net.

1              in     ptr    dns.sar.net.
2              in     ptr    solar.sar.net.
254            in     ptr    gw-solar.sar.net.
```

Figura A-7. Archivo *named.rev* para el Dominio `64.13.200.in-addr.arpa`

Se puede observar que el símbolo @ en el contexto de *named.rev* identifica este registro como el inicio de autoridad (*Start of Authority*) para el dominio `64.13.200.in-addr.arpa` que este está definido en el archivo de arranque *named.boot*. Se puede observar que en archivo named.hosts existen dos nombres apuntando a un a misma dirección, pero en el archivo *named.rev* no se puede configurar esto, únicamente se tiene que colocar un nombre completo válido. Esta es una característica del archivo de nombres, que permite tal flexibilidad, en el archivo de direcciones no se puede hacer lo anterior, sólo apunta una dirección a un nombre.

Si uno quiere configurar los dominios inversos que se observan en el archivo de nombres y se tiene la delegación de esos dominios, se tiene que crear los archivos previamente determinados en el archivo *named.boot,* por ejemplo para el dominio `68.13.200.in-addr.arpa`, se crea un archivo llamado *200.13.68.rev* y para el dominio `76.13.200.in-addr.arpa` se le llamará `200.13.76.rev`:

```
;Archivo 200.13.68.rev
@     in    soa   dns.sar.net armando.sar.net   (
                        9603.0101   ; yymm.dd[0-9] [0-9]serial
                        3600        ; refresh
                        3600        ; retry
                        604800      ; expire
                        172800   )  ; minimum

                        in    ns    dns.sar.net.
                        in    ns    ns.mci.net.


1           in    ptr   slip1.sar.net.
2           in    ptr   slip2.sar.net.
3           in    ptr   slip3.sar.net.
254         in    ptr   cs.sar.net.
```

Figura A-8. Archivo de Dominio Inverso para `68.13.200.in-addr.arpa`

```
;Archivo 200.13.76.rev
@     in    soa   dns.sar.net armando.sar.net   (
                        9603.0101   ; yymm.dd[0-9] [0-9]serial
                        3600        ; refresh
                        3600        ; retry
                        604800      ; expire
                        172800   )  ; minimum

                        in    ns    dns.sar.net.
                        in    ns    ns.mci.net.

1           in    ptr   correo.cotz.sar.net.
25          in    ptr   quija.cotz.sar.net.
254         in    ptr   gw-cotz.cotz.sar.net.
```

Figura A-9. Archivo de Dominio Inverso para `76.13.200.in-addr.arpa`

Los dominios mostrados en las figuras anteriores están configurados previamente en *named.boot* así que el signo @ para el tipo SOA representa la zona del dominio inverso para cada archivo.

Existen muchas aplicaciones como tcpwappers, firewalls, ftp anónimos, telnet seguros, etc. que si determinan un nombre completo haciendo la resolución inversa y por nombre, para determinar que el nombre es válido, de lo contrario determinan que el *host* no es confiable por no pertenecer a un dominio específico.

A.3.1 CONFIGURACIÓN DE UN SERVIDOR DE SÓLO CACHE

Para instalar un *servidor de solo cache*, se requiere tener en la configuración los siguientes archivos: *named.boot, named.ca, named.loca*. Este tipo de servidor es un servidor no autorizado para un dominio determinado (excepto el dominio que forma con su dirección de *loopback 0.0.127.in-addr.arpa*). Este nombre que tiene de solo cache significa que funciona mejorando las búsquedas y el cache de los datos que se realizan.

La configuración típica para un servidor de sólo cache se muestra en el archivo *named.boot:*

```
;Configurando para un Servidor de Solo Cache
directory   /var/named
primary     0.0.127.in-addr.arpa          named.local
cache       .                        named.ca
```

Figura A-10. Archivo *named.boot* para una Configuración de *Servidor de Sólo Cache*

En el archivo mostrado en la figura anterior se puede observar que existe una configuración básica para funcionar como un *servidor de sólo cache*, no indicando la autoridad sobre algún otro dominio en específico, sólo el de *dirección loopback* que representa la única autoridad sobre el *host* mismo.

Para finalizar la configuración de un *servidor de solo cache* se debe apuntar que un servidor de este tipo puede buscar nombres dentro y fuera de su dominio, como lo realiza un servidor primario o secundario. La diferencia es que en los *servidores de solo cache* cuando buscan un nombre que pertenezca a su dominio, finalizan la tarea de búsqueda preguntando a un servidor primario o secundario de su propio dominio.

A.3.2 CONFIGURACIÓN DEL SERVIDOR MAESTRO AUTORIZADO

Un servidor maestro autorizado es aquel que tiene la autoridad previamente solicitada para resolver nombres y direcciones que le sean solicitados. Un SMA tiene dos variantes: *servidores primarios* y *servidores secundarios.*

CONFIGURACIÓN DE UN SERVIDOR PRIMARIO

Un servidor primario es el que tiene localmente sus bases de información y que en ellas se realizan las modificaciones. En el archivo *named.boot* se configura este tipo de servidor con las directivas que lo hacen ser primario, a continuación se presenta el ejemplo de la configuración que se ha venido trabajando:

```
;Configuración de un servidor primario con un dominio
directory                                /var/named
primary     sar.net                      named.hosts
primary     64.13.200.in-addr.arpa       named.rev
primary     68.13.200.in-addr.arpa       200.13.68.rev
primary     76.13.200.in-addr.arpa       200.13.76.rev
primary     0.0.127.in-addr.arpa         named.local
cache       .                            named.ca
```

Figura A-11. Configuración de un *Servidor Primario* en el *named.boot*

Se puede observar en la figura anterior que existe el comando *primary,* que significa el *host* local tiene la administración del servidor primario del dominio de nombres como de los dominios inversos (declarados en la tabla): `sar.net`, `64.13.200.in-addr.arpa`, `68.13.200.in-addr.arpa`, `76.13.200.in-addr.arpa` y bajo el directorio */var/named* se almacenarán los archivos de administración.

Pero este archivo de configuración puede quedar de diferente forma si se cambian los nombres de las bases de datos:

```
;Configuración de un servidor primario con un dominio
directory                                /var/named
primary     sar.net                      sar.net
primary     64.13.200.in-addr.arpa       200.13.64.rev
primary     68.13.200.in-addr.arpa       200.13.68.rev
primary     76.13.200.in-addr.arpa       200.13.76.rev
primary     0.0.127.in-addr.arpa         named.local
cache       .                            named.ca
```

Figura A-12. Configuración Cambiando Nombres Convencionales

A.3.2 CONFIGURACIÓN DEL SERVIDOR MAESTRO AUTORIZADO

Un servidor maestro autorizado es aquel que tiene la autoridad previamente solicitada para resolver nombres y direcciones que le sean solicitados. Un SMA tiene dos variantes: *servidores primarios y servidores secundarios.*

CONFIGURACIÓN DE UN SERVIDOR PRIMARIO

Un servidor primario es el que tiene localmente sus bases de información y que en ellas se realizan las modificaciones. En el archivo *named.boot* se configura este tipo de servidor con las directivas que lo hacen ser primario, a continuación se presenta el ejemplo de la configuración que se ha venido trabajando:

```
;Configuración de un servidor primario con un dominio
directory                                   /var/named
primary     sar.net                         named.hosts
primary     64.13.200.in-addr.arpa          named.rev
primary     68.13.200.in-addr.arpa          200.13.68.rev
primary     76.13.200.in-addr.arpa          200.13.76.rev
primary     0.0.127.in-addr.arpa            named.local
cache       .                               named.ca
```

Figura A-11. Configuración de un *Servidor Primario* en el *named.boot*

Se puede observar en la figura anterior que existe el comando *primary*, que significa el *host* local tiene la administración del servidor primario del dominio de nombres como de los dominios inversos (declarados en la tabla): `sar.net`, `64.13.200.in-addr.arpa`, `68.13.200.in-addr.arpa`, `76.13.200.in-addr.arpa` y bajo el directorio */var/named* se almacenarán los archivos de administración.

Pero este archivo de configuración puede quedar de diferente forma si se cambian los nombres de las bases de datos:

```
;Configuración de un servidor primario con un dominio
directory                                   /var/named
primary     sar.net                         sar.net
primary     64.13.200.in-addr.arpa          200.13.64.rev
primary     68.13.200.in-addr.arpa          200.13.68.rev
primary     76.13.200.in-addr.arpa          200.13.76.rev
primary     0.0.127.in-addr.arpa            named.local
cache       .                               named.ca
```

Figura A-12. Configuración Cambiando Nombres Convencionales

Lo único que se hizo en la configuración de la tabla mostrada anteriormente fue cambiar el nombre de los archivos, obviamente, se tiene que cambiar físicamente los nombres de los archivos de datos.

CONFIGURACIÓN DE UN SERVIDOR SECUNDARIO

Un servidor secundario es un *host* que mantendrá las copias de la información de un servidor primario autorizado, mediante el mecanismo de *zonas de transferencia*. Un servidor secundario es importante configurarlo y darle autoridad de responder, dado que si el servidor primario no puede responder, el servidor secundario autorizado lo puede hacer.

A continuación se presenta una configuración de un servidor secundario en el archivo *named.boot*, se configura en otro *host* para mantener duplicados de la información del servidor primario:

```
; Configuración típica de un servidor secundario del dominio sar.net
directory        /var/named
secondary        sar.net                200.13.64.1       sar.net.bak
secondary    64.13.200.in-addr.arpa     200.13.64.1       200.13.64.rev.bak
secondary    68.13.200.in-addr.arpa     200.13.64.1       200.13.68.rev.bak
secondary    76.13.200.in-addr.arpa     200.13.64.1       200.13.76.rev.bak
primary      0.0.127.in-addr.arpa            named.local
cache        .                               named.ca
```

Figura A-13. Configuración de un Servidor Secundario

En la figura anterior la directiva *secondary*, hace que el host local se convierta en un servidor secundario del dominio. En ese mismo comando se determina la dirección IP del servidor primario para el dominio de nombre como el (los) dominio (s) inverso(s).

Es importante mencionar que los archivos con terminación *bak* (*de backup*) están sin crearse hasta que ocurre un zona de transferencia entre el *host* que tiene el servidor secundario con el servidor primario.

RESUMEN

Para finalizar, un servidor maestro autorizado puede combinarse siendo primario, secundario o de solo cache. También es importante decir que se puede tener en un dominio primario más de un dominio de nombres como dominios inversos, pero, al menos cuando son servidores primarios o secundarios, el *host* que representa el tipo SOA tiene que estar registrado en el DNS. El *host* que está sirviendo de servidor puede configurarse como cliente del DNS, es decir, agregando a cada servidor el archivo */etc/resolv.conf*.

Podemos mencionar que las configuraciones son:

- Configuraciones de servidores Multidominios
  Cuando en un servidor primario existe más de un dominio por nombre o dominio inverso.
- Configuraciones de servidores Combinados
  Cuando en un servidor tiene configuración de servidor primario, secundario y sólo cache.
- Configuraciones donde pueden ser clientes
  Cuando se agrega el archivo */etc/resolv.conf*.

Se puede analizar que para que un servidor maestro autorizado (*primario, secundario*) funcione correctamente, se tiene que configurar como *servidor de sólo cache*, para aprovechar las características que este servidor proporciona.

A.4 ADMINISTRANDO EL SERVIDOR (NAMED)

Se han visto las configuraciones para servidores de sólo cache, primario y secundario, pero falta mencionar algo que es muy importante para el funcionamiento de resolución de los servidores que es; la ejecución del *daemon* o programa *named*.

El *named* es el servidor que interpretará la configuración del archivo */etc/named.boot* y hará válida dicha configuración.

```
# named
```

Figura A-14. Iniciando el Servidor *named*

Cuando se realiza una modificación al archivo *named.boot* o a alguna base de datos, se tiene que modificar el registro SOA en su número serial y algo importante es volver a ejecutar el servidor named. En algunos sistemas es necesario ver el proceso del servidor y recargar el servidor, corriéndolo con el mismo número de proceso, con la finalidad de que el servidor tome los cambios, esto es:

```
# ps -fea | grep named
named       50          hui   ?           ?     R
# kill -HUP 50
#
```

Figura A-15. Restablecimiento del Servidor *named*

En otros sistemas el número de proceso se almacena en un archivo */etc/named.pid*, lo cual para recargar el servidor lo que se hace es:

```
#kill -HUP `cat /etc/named.pid`
#
```

Figura A-16. Reinicialización del Servidor *named*

Para ejecutar y recargar el servidor es necesario que alguien con la clave de *superusuario* en el sistema lo realice, o bien, se puede configurar el ambiente para que un usuario especial haga esta tarea.

ANEXO B. SOLICITUDES DE DOMINIOS

B.1 DOMINIOS

La dinámica en que Internet se desarrolla es siguiendo la lógica de la referencia a un nombre que indica posición y dirección. Tal nombre pertenece a lo que se maneja con el concepto de *dominio*.

Los dominios en Internet pueden ser por nombres o por dirección inversa. Genéricamente se conocen por dominios a los dos. Los dominios por nombre están formados por etiquetas y separados por puntos. Los dominios de dirección inversa, están formados por la dirección de red en forma inversa y un dominio genérico conocido como *in-addr.arpa*.

Para finalizar, es importante mencionar que tener un dominio no solo basta para permanecer dentro de Internet, es necesario complementarlo con direcciones IP (Internet Protocol) que formarán los dominios inversos y colocarlo en un servidor de nombres que brinde la resolución de los nombres para dichos dominios.

La administración actual de Internet se hace a través de INTERNIC que es un asociación que coordina, administra y tiene autoridad sobre la red de redes. A través de este organismo, se hacen diversos trámites coordinados con los proveedores de acceso al *"backbone"* a Internet. Básicamente dentro de INTERNIC existen los servicios de administrar bases de datos formadas por información referente a los dominios administrados en el mundo, las delegaciones de direcciones, la información de contactos técnicos y administrativos, información relacionada a las creaciones y propuestas tecnológicas en Internet (RFCs).

En sí, INTERNIC es un sistema de información global de Internet, donde tienen diversos servicios para la comunidad. Ahora, en INTERNIC se cobra un costo de solicitud y administración por dominio; esto también ocurre en otros países, en el caso de México todavía no se cobra, pero está pronto de implantarse.

La administración de cada país se ha realizado a través de una organización que INTERNIC designa. INTERNIC provee los servicios de registro para los dominios más altos en el nivel jerárquico: **.COM**, **.EDU**, **.NET**, **.ORG** y **.NET**, pero

otro tipo de solicitudes de registro de dominio deben dirigirse a las organizaciones autorizadas por INTERNIC.

Por ejemplo para el registro del dominio **.US** que se refiere al registro para organizaciones dentro del dominio de .US bajo la Autoridad de Números Asignados en Internet *(IANA, the Internet Assigned Numbers Authority).*

También para el registro **.CA** que concierne al dominio de Canadá (tanto registro de dominios como para direcciones IP canadienses).

En RIPE NCC se realizan las actividades de registro de servicios en Europa.

En APNIC (Asia Pacific Network Information Center) se provee registro de solicitudes de servicios en la región del Pacífico de Asia.

En México, existe un NIC-México, conocido como el NIC MEXICANO que tiene el papel de informar, administrar, asignar y mantener a los dominios por nombres e inversos que se solicitan.

Para accesar a INTERNIC a través de *http://www.internic.net.*

Para accesar al NIC-México a través de http://www.nic.mx.

B.2 REPRESENTACIÓN DE INTERNET

Dentro de Internet se tienen códigos que describen a los países (basado en el ISO* 3166), representando al dominio padre por cada país. INTERNIC es la representación máxima dentro de Estados Unidos y en el mundo (donde se fundó Internet) y es por la cual, tiene autoridad sobre la red y además tiene los dominios base conocidos como:

DOMINIO	REPRESENTACION
org	Organizaciones no gubernamental o sin fines de lucro.
com	Relacionado con el giro comercial.
net	Proveedores de acceso o conectividad.
edu	Instituciones educativas.
arpa	Dominio especial de direcciones inversas.
mil	Organizaciones militares.

Tabla B-1. Representación de Dominios en Internet

Basados en las representaciones de los dominios mencionados, INTERNIC utilizó los códigos del estándar de descripción de países con el cual identifica a cada uno para comenzar la representación de zonas o áreas de actividad, tal como, áreas comerciales, de conectividad, educación, etc. Por ejemplo, utilizar para México el **.MX** y tener áreas como: **.COM.MX**, **.NET.MX**, **ORG.MX**, etc.

A continuación se presenta una tabla donde se listan los códigos de paises (etiquetas para identificar al país) que se encuentran dentro de Internet.

*ISO 3166, es un estándar que define códigos para la representación de los países del mundo de la Organización Internacional para la Estandarización (International Organization for Standardization).

DOMINIO	PAIS	DOMINIO	PAIS
ad	Andorra	cl	Chile
ae	United Arab Emirates	cm	Cameroon
af	Afghanistan	cn	China
ag	Antigua and Barbuda	co	Colombia
ai	Anguilla	cr	Costa Rica
al	Albania	cs	Czechoslovakia (former)
am	Armenia	cu	Cuba
an	Netherlands Antilles	cv	Cape Verde
ao	Angola	cx	Christmas Island
aq	Antarctica	cy	Cyprus
ar	Argentina	cz	Czech Republic
as	American Samoa	de	Germany
at	Austria	dj	Djibouti
au	Australia	dk	Denmark
aw	Aruba	dm	Dominica
az	Azerbaijan	do	Dominican Republic
ba	Bosnia and Herzegovina	dz	Algeria
bb	Barbados	ec	Ecuador
bd	Bangladesh	ee	Estonia
be	Belgium	eg	Egypt
bf	Burkina Faso	eh	Western Sahara
bg	Bulgaria	er	Eritrea
bh	Bahrain	es	España
bi	Burundi	et	Ethiopia
bj	Benin	fi	Finland
bm	Bermuda	fj	Fiji
bn	Brunei Darussalam	fk	Falkland Islands (Malvinas)
bo	Bolivia	fm	Micronesia
br	Brazil	fo	Faroe Islands
bs	Bahamas	fr	France
bt	Bhutan	fx	France, Metropolitan
bv	Bouvet Island	ga	Gabon
bw	Botswana	gb	Great Britain (UK)
by	Belarus	*gd*	*Grenada*
bz	Belize	*ge*	*Georgia*
ca	Canada	*gf*	*French Guiana*
cc	Cocos (Keeling) Islands	*gh*	*Ghana*
cf	Central African Republic	*gi*	*Gibraltar*
cg	Congo	*gl*	*Greenland*
ch	Switzerland	*gm*	*Gambia*
ci	Cote D'Ivoire (Ivory Coast)	*gn*	*Guinea*
ck	Cook Islands	*gp*	*Guadeloupe*

Tabla B-2. Relación de Dominios por Países

DOMINIO	PAIS	DOMINIO	PAIS
gd	Grenada	km	Comoros
ge	Georgia	kn	Saint Kitts and Nevis
gf	French Guiana	kp	Korea (North)
gh	Ghana	kr	Korea (South)
gi	Gibraltar	kw	Kuwait
gl	Greenland	ky	Cayman Islands
gm	Gambia	kz	Kazakhstan
gn	Guinea	la	Laos
gp	Guadeloupe	lb	Lebanon
gq	Equatorial Guinea	lc	Saint Lucia
gr	Greece	li	Liechtenstein
gs	S. Georgia and S. Sandwich Isls.	lk	Sri Lanka
gt	Guatemala	lr	Liberia
gu	Guam	ls	Lesotho
gw	Guinea-Bissau	lt	Lithuania
gy	Guyana	lu	Luxembourg
hk	Hong Kong	lv	Latvia
hm	Heard and McDonald Islands	ly	Libya
hn	Honduras	ma	Morocco
hr	Croatia (Hrvatska)	mc	Monaco
ht	Haiti	md	Moldova
hu	Hungary	mg	Madagascar
id	Indonesia	mh	Marshall Islands
ie	Ireland	mk	Macedonia
il	Israel	ml	Mali
in	India	mm	Myanmar
io	British Indian Ocean T.	mn	Mongolia
iq	Iraq	mo	Macau
ir	Iran	mp	Northern Mariana Islands
is	Iceland	mq	Martinique
it	Italy	mr	Mauritania
jm	Jamaica	ms	Montserrat
jo	Jordan	mt	Malta
jp	Japan	mu	Mauritius
ke	Kenya	mv	Maldives
kg	Kyrgyzstan	mw	Malawi
kh	Cambodia	mx	**México**
ki	Kiribati	my	Malaysia

DOMINIO	PAIS	DOMINIO	PAIS
mz	Mozambique	sh	St. Helena
na	Namibia	si	Slovenia
nc	New Caledonia	sl	Sierra Leone
ne	Niger	sm	San Marino
nf	Norfolk Island	sn	Senegal
ng	Nigeria	so	Somalia
nl	Nicaragua	sr	Suriname
nl	Netherlands	st	Sao Tome and Principe
no	Norway	su	USSR (former)
np	Nepal	sv	El Salvador
nr	Nauru	sy	Syria
nt	Neutral Zone	sz	Swaziland
nu	Niue	tc	Turks and Caicos Islands
nz	New Zealand (Aotearoa)	td	Chad
om	Oman	tf	French Southern Territories
pa	Panama	tg	Togo
pe	Perú	th	Thailand
pf	French Polynesia	tj	Tajikistan
pg	Papua New Guinea	tk	Tokelau
ph	Philippines	tm	Turkmenistan
pk	Pakistán	tn	Tunisia
pl	Poland	to	Tonga
pm	St. Pierre and Miquelon	tp	East Timor
pn	Pitcairn	tr	Turkey
pr	Puerto Rico	tt	Trinidad and Tobago
pt	Portugal	tv	Tuvalu
pw	Palau	tw	Taiwan
py	Paraguay	tz	Tanzania
qa	Qatar	ua	Ukraine
re	Reunion	ug	Uganda
ro	Romania	uk	United Kingdom
ru	Russian Federation	um	US Minor Outlying Islands
rw	Rwanda	us	United States
sa	Saudi Arabia	uy	Uruguay
sb	Solomon Islands	uz	Uzbekistan
sc	Seychelles	va	Vatican City State (Holy See)
sd	Sudan	vc	Saint Vincent and the Grenadines
se	Sweden	ve	Venezuela
sg	Singapore	vg	Virgin Islands (British)

DOMINIO	PAIS
vi	Virgin Islands (U.S.)
vn	Viet Nam
vu	Vanuatu
wf	Wallis and Futuna Islands
ws	Samoa
ye	Yemen
yt	Mayotte
yu	Yugoslavia
za	South Africa
zm	Zambia
zr	Zaire
zw	Zimbabwe
com	US Commercial
edu	US Educational
gov	US Government
int	International
mil	US Military
net	Network
org	Non-Profit Organization
arpa	Old style Arpanet
nato	Nato field

B.3 POLÍTICAS PARA DOMINIOS

Las diversas representaciones de Internet en los países están definidos por INTERNIC. En México, el NIC-México es la máxima representación ha establecido una serie de políticas para la obtención de dominios.

En esta sección del anexo, se presentan las políticas que presenta el NIC MX ante administradores de redes públicas y proveedores de acceso a Internet. Las políticas están basadas propiamente en algunas características obtenidas de INTERNIC y otras han sido escritas para satisfacer algunas necesidades de las circunstancias de las redes nacionales.

La razón por la cual se diseñan las políticas es para controlar el flujo de solicitudes y peticiones para creación de dominios, además de los que no tienen sentido y más aún tienen nombres denigrantes ante la sociedad.

Cabe destacar que el NIC de México es manejado por el ITESM de Monterrey y tiene la autoridad de manejar el DNS nacional. Tiene entonces la autoridad de decidir si un dominio se acepta o se niega a un usuario responsable de una red que se desee conectar a Internet.

A continuación se presentarán las políticas para la obtención de un dominio (por nombre) y las políticas de asignación de direcciones IP que está relacionado con el dominio inverso. Es importante decir, que estas políticas están establecidas con toda la autoridad que INTERNIC delega a los países.

POLÍTICAS PARA OBTENCIÓN DEL DOMINIO

[ftp://ftp.nic.mx/pub/templates/politicas.txt 5k]
[http://www.nic.mx/NIC/ip-reassign.html]

Politicas para registro
de Nombres de Dominio NIC-Mexico

+----------+

A continuacion se describen las politicas para registrar nombres de dominio ante NIC-Mexico bajo .MX. Se da por hecho que se conocen las politicas de registro de dominio de INTERNIC [RFC 1591].

1. Administracion de Dominios.

El NIC-Mexico (auspiciado por el ITESM, Campus Monterrey, a traves del Departamento de Telecomunicaciones y Redes en la Direccion de Informatica), es el responsable de la coordinacion y el manejo del DNS nacional, y especialmente en la delegacion de dominios bajo .MX, asi como del registro de direcciones inversas para los dominios que tenga asignados para ello.

Todas las solicitudes de registro de dominio deben ser enviadasa NIC-Mexico (en HOSTMASTER@NIC.MX).

2. Responsabilidades de NIC-Mexico

Realizar un trabajo satisfactorio en el mantenimiento del DNS para el dominio .MX

Esto es, que la asignacion de nombres de dominios, delegacion de subdominios y operacion de nameservers deben ser hechos con un trabajo competitivo. Asi como el mantenimiento de las bases de datos con eficiencia, para un tiempo de respuesta optimo.

Ser equitativo con todos los solicitantes de nombres de dominios.

Esto significa que las mismas reglas y politicas se aplican a todas las solicitudes, las cuales deben procesarse en un esquema no discriminatorio sin importar el tipo de cliente (comercial, educativo, etc.). [RFC-1591].

3. Responsabilidades del solicitante.

Es responsabilidad del solicitante de registro de dominio:

- o Proponer un nombre de dominio de acuerdo informacion del inciso 4 y 5 de estas politicas.
- o Mantener informado a NIC-Mexico de cualquier modificacion o actualizacion sobre dominios previamente registrados.
- o Asegurarse del funcionamiento del servidor de nombres para su dominio desde el momento de solicitar este y durante su operacion.

o Registrar el dominio inverso ante la autoridad del dominio.

o Identificar quien es la autoridad de su dominio.

4. Identificacion del dominio padre.

La siguiente es la estructura en que estan identificados los dominios bajo .MX

.mx	Instituciones de educacion o investigacion.
.com.mx	Entidades comerciales.
.org.mx	Asociaciaciones no lucrativas.
.gob.mx	Asociaciones gubernamentales.
.net.mx	Proveedores de servicios de red.

Para resgistrar dominios bajo GOB.MX es requisito anexar a la solicitud de registro de dominio, una carta con membrete de la oficina de gobierno que se desea dar de alta.

Para acelerar el tramite de estos dominios, esta carta puede mandarse por via fax con atencion a:

Ing. Oscar Robles
Dpto. Telecomunicaciones y Redes

Al siguiente FAX: 52 (8) 3 28 4208. Y el original via correo postal

Ing. Oscar Robles
Dpto. Telecomunicaciones y Redes
ITESM, Campus Monterrey
Ave. Eugenio Garza Sada #2501
C.P. 64849

5. Nombres y Marcas Registradas.

El registrar un nombre de dominio no significa registrar una marca. Es requisito del solicitante asegurarse de que no esta violando ninguna marca registrada.

En caso de una disputa entre solicitantes por los derechos de un nombre en particular, la autoridad que registra el nombre (NIC-Mexico) no debera tener responsabilidad alguna, mas que proveer informacion a ambas partes.

6. Nombres de Dominios.

a.- La longitud del nombre de dominio no debera exceder los 12 caracteres.

b.- Para evitar competencia desleal y/o problemas posteriores con marcas registradas, NO deberan registrarse nombres de dominios con alguna de las siguientes caracteristicas:

- Nombres Geograficos *
  - mexico.com.mx
  - monterrey.mx
  - veracruz.mx
  - mty.com.mx

- Nombres genericos
  - libros.com.mx
  - papelerias.com.mx
  - escuela.mx

* En caso de que se solicite registrar un dominio bajo GOB.MX, los nombres geograficos son aceptados.

Este apartado no solo hace referencia a los dominios .COM.MX, tambien estan incluidos: .MX, .NET.MX, .ORG.MX, .GOB.MX.

c.- El nombre de dominio no debe hacer referencia a aspectos oficiales sin tener la autorizacion para serlo.

Por ejemplo.
- gobweb.com.mx
- webgob.mx

Estos dominios hacen referencia explicita a la pagina de WEB del Gobierno Mexicano.

* ~ *

Mayor informacion:
---
help@nic.mx o ayuda@nic.mx
FAQ NIC-Mexico
---
http://www.nic.mx/NIC/faq.html
Lista de INFORMACION de NIC-Mexico

Mandar email a majordomo@nic.mx con el siguiente mensaje:
subscribe info
Lista de DISCUSION de NIC-Mexico

Mandar email a majordomo@nic.mx con el siguiente mensaje:
subscribe dns

~

POLÍTICAS PARA ASIGNACIÓN DE CLASES DE DIRECCIONES IP

Politicas para Asignacion
de Clases C y Direcciones IP NIC-Mexico
+--+

A continuacion se describen las politicas para asignacion de clases C's y direcciones IP del NIC-Mexico.

1. Delegacion de Direcciones IP

El NIC-Mexico, como organismo coordinador del crecimiento del Internet en Mexico, tiene a disponibles bloques de direcciones IP (CIDR), es decir bloques de clases C's para reasignar principalmente a Proveedores de acceso en Mexico, quienes a su vez son responsables de la correcta administracion y delegacion de direcciones que les sean proveidas por el NIC-Mexico.

2. Responsabilidades de NIC-Mexico

El NIC-Mexico se hace responsable de:

a.- Atender a todas las solicitudes de manera equitativa, esto significa que las mismas reglas y politicas se aplican a todas las solicitudes, las cuales deben procesarse en un esquema no discriminatorio sin importar el tipo de cliente

b.- Asignar Clases C's a los Proveedores de acceso que asi lo requieran.

c.- Mandar las formas de SWIP (Shared WHOIS Project) al INTERNIC para la correcta reasignacion de las Clases C.

3. Responsabilidades del solicitante.

El solicitante (proveedor de acceso a internet) se hace responsable:

a.- De la correcta administracion y delegacion de direcciones IP a sus clientes, estableciendo politicas y reglas de asignacion que hagan mas eficiente el uso de direcciones IP.

b.- Asimismo, debera informar al NIC-Mexico de cualquier cambio en la administracion del bloque de IP asignado (clase C).

4. Solicitud de Clases C

Para solicitar una Clase C o conjunto de ellas (CIDR), es necesario llenar las formas de registro de IP del NIC-Mexico:
http://www.nic.mx/cgi-bin/ips.pl

En el caso de solicitar mas de una Clase C, es indispensable incluir una justificacion. esta justificacion debera incluir por lo menos:

- Tipos de servicios que se ofrecen
- Bloques de direcciones previamente asignados
- Tipos de asignacion de direcciones
- Mascaras utilizadas
- Planeacion de la topologia o uso de esas Clases C's

Esta informacion debera enviarse junto con la forma de la pagina del WEB, en su defecto via FAX: (8) 3 28 4208.

O bien, por correo postal a:

Ing. Oscar Robles
Dpto. Telecomunicaciones y Redes
ITESM, Campus Monterrey
Ave. Eugenio Garza Sada #2501
C.P. 64849

* ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ *

Mayor informacion:
---
help@nic.mx o ayuda@nic.mx
FAQ NIC-Mexico
---
http://www.nic.mx/NIC/faq.html
Lista de INFORMACION de NIC-Mexico

Mandar email a majordomo@nic.mx con el siguiente mensaje:
subscribe info
Lista de DISCUSION de NIC-Mexico

Mandar email a majordomo@nic.mx con el siguiente mensaje:
subscribe dns

* ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ * ~ *

B.4 SOLICITUDES DEL NIC Y NIC-MÉXICO

Para obtener los servicios de dominio tanto por nombres como inverso, es necesario contemplar la necesidad de enviar solicitudes de dominios tanto para nombres como inversos. Cumpliendo previamente con las políticas de uso del NIC de INTERNIC y el NIC-México se procede a enviar las siguientes formas a través de correo electrónico.

Se presentan las solicitudes en español y algunas partes de instrucción en inglés debido a que el NIC-México es intermediario del trámite de solicitud de dominio inverso.

SOLICITUD DE DOMINIO

```
 URL ftp://ftp.nic.mx/pub/templates/domain-template.txt ]
[ URL http://www.nic.mx/cgi-bin/dominio.pl ]
[ URL http://www.nic.mx/cgi-bin/domain.pl ]

---------------------------------------------------------------

                        NIC-MEXICO

        Domain Names Delegation for .MX Domain
---------------------------------------------------------------
---------------------------------------------------------------

0.  Movement.........:
---------------------------------------------------------------
1.  Purpouse.........:

2.  Domain...........:

3a. Organization Name.......:
3b. Org. Address............:

4.  Operational Date........:

Administrative Contact
5a. NIC Handle.......:
5b. Name.............:
5c. Organization.....:
5d. Postal Addressi..:
5e. Phone............:
5f. E-Mail...........:

Technical Contact
6a. NIC Handle.......:
6b. Name.............:
6c. Organization.....:
6d. Postal Address...:
6e. Phone............:
6f. E-Mail...........:

Primary Server:
7a. Hostanme:
7b. NetAdress:
7c. Hardware:
7d. Software:

Secondary Server:
8a. Hostname:
8b. NetAdress:
8c. Hardware:
8d. Software:

9a. Hostname:
9b. NetAdress:
```

```
9c. Hardware:
9d. Software:

10a. Hostname:
10b. NetAdress:
10c. Hardware:
10d. Software:

-------------------------------------------------------------

The party requesting registration of this name certifies that, to
her/his
knowledge, the use of this name does not violate trademark or other
statutes.

Registering a domain name does not confer any legal rights to that name
and
any disputes between parties over the rights to use a particular name
are to
be settled between the contending parties using normal legal methods.

(See:
      ftp://ftp.nic.mx/pub/templates/politicas.txt
      and, RFC 1591)
```

SOLICITUD DE DOMINIO INVERSO (IN-ADDR.ARPA)

```
Registration Action Type
  0.  (N)ew (M)odify (D)elete:

  Network Information
  1a.  Network Name.................:
  1b.  Start of Network Block.......:
  1c.  End of Network Block.........:

  2a.  Name of Organization.........:
  2b.  Postal address of Organization:

  Technical Contact
  3a.  NIC Handle (if known).........:
  3b.  Name (Last, First)............:
  3c.  Organization..................:
  3d.  Postal Address................:

  3e.  Phone Number..................:
  3f.  E-Mail Address................:

  Primary Name Server
  4a.  Primary Server Hostname.......:
  4b.  Primary Server Netaddress.....:

  Secondary Name Server(s)
  5a.  Secondary Server Hostname.....:
  5b.  Secondary Server Netaddress...:

  6.   Comments.....................:

------------------------------ cut here ------------------------

                        GENERAL INSTRUCTIONS

    What is an IN-ADDR domain:

    The Internet uses a special domain to support address to name
    mapping, referred to as inverse-addressing (IN-ADDR).

    IN-ADDR domains are represented using the network number in
reverse.
    For example, the IN-ADDR domain for network 123.45.67.0 is
    represented as 67.45.123.IN-ADDR.ARPA.  PLEASE DO NOT LIST YOUR
    NETWORK NUMBER IN REVERSE ON YOUR TEMPLATE.

    Use the above template for registering new IN-ADDR entries, making
changes to existing IN-ADDR records, and removing inverse-address
mapping from the InterNIC database and root servers.
```

This template, and only this template, should be sent via e-mail to:

hostmaster@campus.mty.itesm.mx

Please do not send hardcopy registrations to the InterNIC. Your provider will be able to send e-mail applications for you if you are not connected.

Please do not modify the template.
PLEASE SEND ONLY ONE TEMPLATE PER MESSAGE.

In the Subject of the message, use the words, "NEW IN-ADDR", "MODIFY IN-ADDR", or "REMOVE IN-ADDR" as appropriate.

Section 0 - Registration Action Type

"N" - New IN-ADDR registration.
The letter "N" or the word "New" indicates a NEW registration.

"M" - Modify an existing IN-ADDR registration.

When "M" is selected, the current records will be replaced with the information listed in the template. Please provide a complete list of name servers in the order that they should appear on the record.

If the modification involves first registering a person or name server(s) that are not in the database, the instructions for completing Sections 2, 3, 4 and 5 apply. Use the "WHOIS" database if you are not sure about the current information for a technical point of contact or name server(s).

Changes will be made if it appears to the operator that the modification request has come from an authorized source.
This source could be from a listed contact for the domain, from others in the same organization, from the current provider, or from a new provider that is about to provide support for the network.

"D" - Delete existing IN-ADDRs from record.

The default for this option will be to remove IN-ADDRs from the network number or block entry listed. The host entry will still exist in the global host tables.

Section 1 - Network Record Information

Network Name -

THIS IS NOT YOUR DOMAIN NAME. Please supply the network name listed in the network's WHOIS record.

Start/End of Network Block -

If the network record is a single network, item 1b would be the IP address of the single network and item 1c would be blank. If the network record is a block of networks, item 1b would be the IP address of the start of the network block and item 1c would be the IP address of the end of the network block.

If you received a block of IP addresses from your Internet Service Provider, there may already be domain name servers on the larger block of addresses held by the provider. Please query your Internet Service Provider before submitting a request for inverse addressing.

Section 2 - Name and postal address of Organization

The network is considered to be registered to an organization, even if the "organization" is an individual. If you are an Internet Service Provider submitting this request on behalf of your client, please use the name and postal address of the organization utilizing the IP address(es).

When completing item 2b, place the city, state, and zip code on a separate line. Use a comma to separate the city and state. Do not insert a period following the state abbreviation. For example:

```
Organization address.: Street or PO Box
                       Herndon, VA 22070
```

If the organization is in a country other than the United States, please include the name of the country on the last line by itself. For example:

```
Organization address.: Street or PO Box
                       Montreal, QC H2S 2C8
                       Canada
```

Section 3 - Technical Point of Contact

The technical point of contact is the person who tends to the technical aspects of maintaining the network's name servers. This person should be able to answer any utilization questions the InterNIC may have.

Each person in the InterNIC database is assigned a "handle" - a unique tag consisting of the person's initials and a serial number. This tag is used on records in the database to indicate a point of contact for a domain name, network, name server or other entity. Each person should have only one handle.

IF THE PERSON'S HANDLE IS KNOWN, INSERT JUST THE HANDLE IN ITEM 3a AND LEAVE THE REST OF SECTION 3 BLANK. IF THE PERSON'S

HANDLE IS UNKNOWN OR THE PERSON HAS NEVER BEEN REGISTERED, LEAVE ITEM 3a BLANK. The user's database record will be updated with any new information on the template.

Item 3c refers to the name of the organization with which the technical point of contact is affiliated. Item 3d should be completed following guidelines set forth regarding item 2b.

Section 4 - Primary Name Server

Networks are required to provide at least two independent servers for translating address to name mapping for hosts in the domain.
The servers should be in physically separate locations and on different networks if possible. The servers should be active and responsive to DNS queries BEFORE this application is submitted.
Incomplete information in sections 4 and 5, or inactive servers will result in the return of the registration request.

Neither the name nor the number of a registered name server will be changed as a result of a new IN-ADDR registration. A Modify registration request must be sent to change either of these values.

Please provide the fully-qualified name of the machine that is to be the name server; for example: "machine.domainame.com" not just "machine" or just "domainame.com" Many reverse-authentication programs will not search for the nameserver if only the domain name is listed.

It is suggested that the fourth octet of an IP address of a server should be neither 0 nor 255. The remaining 254 numbers in the fourth octet of the IP address are valid.

Section 5 - Secondary Name Server(s)

The same procedures for specifying primary servers apply as to secondary servers. If several secondary servers are required, copy Section 5 as many times as needed. DO NOT RENUMBER OR CHANGE THE COPIED SECTION. A maximum of six domain name servers may be added to a network record.

Section 6 - Comments

Please utilize Section 6 for all comments and more detailed updates not covered by template Sections 0 through 5.

B.5 COMENTARIOS SOBRE REGISTRO DE DOMINIOS

Para poder accesar a Internet y utilizar al máximo sus servicios, es de vital importancia que el sitio que se va a conectar obtenga un dominio de nombre para que pueda reconocerse dentro del mundo de la red de redes más grande del mundo. Por lo que el administrador de una red tiene que conocer la existencia de las políticas que existen en Internet, así como la existencia de las solicitudes de los dominios y asignación de direcciones para así anunciar los dominios inversos.

Las formas de solicitud son excluyentes, pero, al momento en que se desea trabajar en Internet, se necesitan complementar. Si uno obtiene un dominio por nombre, trabajar en Internet se hace un poco aislado al no contar con direcciones IP propias (proporcionadas por el proveedor de servicios de Internet), pero, realmente también con direcciones IP y sin dominio, no puedo utilizar todo los servicios que Internet proporciona.

Así que el objetivo que el administrador debe perseguir es el hecho de tener resueltos estos requisitos llenando las solicitudes vistas en este anexo, para hacer funcionar eficientemente la red que se desee conectar a Internet.

Teniendo un sitio con un servidor de nombres bien configurado, se obtiene los servicios que Internet proporciona eficientemente como, correo electrónico, acceso a servidores WWW, ftp anónimo, conexiones remotas, etc. Todos estos servicios funcionan mejor con un DNS bien configurado y además anunciado y reconocido por todo el mundo.

Hoy en día, existe un nuevo mecanismo para tramitar dominios en Internic (en México no está implantado) que es el de solicitar una contraseña a Internic con la finalidad de que cuando se realicen trámites de dominios o petición de direcciones IP todas las transacciones s se lleven a cabo de forma segura. Este nuevo método encripta una contraseña y se tiene que manejar a través de correo electrónico, este nuevo método se conoce como *Guardian*.

En muchos casos los "ISP" (*Internet Service Provider*) o proveedores de acceso a Internet pueden hacer las solicitudes de registro de los dominios llenando las formas de DNS o enviando la contraseña adoptando el nuevo método de tramitación tanto para dominio de nombre como dominio inverso para las instituciones que se conectan.

ANEXO C. MANUAL DE INSTALACIÓN DEL SADNS

C.1 INTRODUCCIÓN

Se presenta la forma de instalar el SADNS(V.1) en una estación de trabajo UNIX SGI Irix 5.3 o SUN Solaris 2.X. Este manual tiene como finalidad poner en funcionamiento del SADNS como herramienta de los usuarios de Internet.

C.2 REQUERIMIENTOS

- Servidor HTTPD (NCSA 1.3 o 1.5.X)
- Navegador Netscape 1.0 o mayor.
- Sistema Operativo UNIX (Solaris 5.X o Irix 5.X)
- SGI Indy, Sparc 20 SUN, Classic SUN
- Servidor de nombres (named o in.named)
- Software de SADNS (ftp://ftp.sar.net/dns/sadns.tar.Z)

C.3 CREANDO EL ÁREA DE INSTALACIÓN DEL SADNS

En el sistema donde se vaya a instalar el sofware SADNS, se requiere para mayor facilidad y utilización del mismo los siguientes pasos:

1. Creación de la clave en el sistema sadns. Puede hacerse utilizando el programa de administración de cada sistema *toolchest para SGI o bien, admintool para SUN*. Es importante colocarle el ambiente del sistema **csh.**

```
sadns: :x:1111:1000:SADNS:/usr/people/sadns:/bin/csh
```

2. Con password * o NP (No Password), según sea el sistema operativo (en los soportados existe el archivo */etc/shadow*, para lograr que la clave creada sea únicamente para área donde residan los programas y configuración del SADNS (no es una clave de un usuario común).

```
sadns:*::::::: (En SGI)
sadns:NP::::::: (En SUN)
```

3. Se crea un grupo **sadns** en el archivo */etc/group* con el número 1000 y quedando dela forma siguiente:

```
sys::0:root,bin,sys,adm
root::0:root
daemon::1:root,daemon
bin::2:root,bin,daemon
adm::3:root,adm,daemon
mail::4:root
uucp::5:uucp
rje::8:
lp:*:9:
nuucp::10:nuucp
user::20:
other::995:
demos:*:997:
guest:*:998:
sadns::1000:sadns
```

4. Cambie los permisos a la clave de **sadns** en forma manual o automática de su administrador de usuarios del sistema.

```
%chmod 700 USUARIOS/sadns
%
%chgrp sadns USUARIOS/sadns
%chgrp sadns USUARIOS/sadns
%
```

Donde **USUARIOS** es el área de usuarios del sistema UNIX.

C.4 OBTENIENDO EL SERVIDOR DE WEB (HTTPD)

El SADNS es un sistema que utiliza la interface gráfica que provee el navegador del Web, por lo que necesitamos un servidor de WWW que nos permita obtener las siguientes características: interface gráfica, autentificación y que utilice lenguaje HTML, por lo que se decidió utilizar el servidor de httpd de NCSA (National Center for Supercomputing Applications). Para realizar la instalación del servidor en el área del SADNS procedemos a realizar lo siguiente: (Esto es si se encuentra trabajando en una SGI)

1. Desde el área de trabajo /usr/people/sadns (HOME de la clave), se tiene que hacer un **ftp** al servidor de **ftp anónimo** al NCSA (*ftp.ncsa.uiuc.edu*) bajo el subdirectorio `/Web/httpd/Unix/ncsa_httpd/httpd_1.5` para obtener el software del servidor de httpd.

```
chajul 6% pwd
/usr/people/sadns
chajul 7% ftp ftp.ncsa.uiuc.edu  Haciendo la conexión via ftp
Connected to ftp.ncsa.uiuc.edu.
220 idunno FTP server (Version wu-2.4(25) Thu Aug 25 13:14:21 CDT 1994)
ready.
Name (ftp.ncsa.uiuc.edu:armando): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230-
230-Welcome to NCSA's new anonymous FTP server! I hope you find what
you
are
230-  looking for. If you have any technical problems with the server,
230-  please e-mail to ftpadmin@ncsa.uiuc.edu. For other questions
regarding
230-  NCSA software tools, please e-mail softdev@ncsa.uiuc.edu.
230-
230-You are user # 22 of an allowed 130 users.
230-
230-Please read the file README
230-  it was last modified on Tue Jan  3 18:54:35 1995 - 543 days ago
230-Please read the file README.FIRST
230-  it was last modified on Thu Jan 12 17:53:58 1995 - 534 days ago
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /Web/httpd/Unix/ncsa_httpd/httpd_1.5
250-Please read the file README
250-  it was last modified on Sat Apr  6 14:27:34 1996 - 84 days ago
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 20750
drwxr-xr-x   2 19056    wsstaff     2048 Apr  5 18:30 .
drwxr-xr-x   8 12873    wheel       2048 Nov 10  1995 ..
-rw-rw-r--   1 19056    wsstaff     5232 Apr  6 15:27 README
-rwxr-xr-x   1 19056    wsstaff   186169 Apr  5 18:29 httpd_1.5.1-aix3.2.5.Z
-rwxr-xr-x   1 19056    wsstaff   396325 Apr  5 18:29 httpd_1.5.1-bsdi2.1.Z
-rw-r--r--   1 19056    wsstaff  1271171 Apr  5 18:29 httpd_1.5.1-bsdi2.1.tar.Z
-rwxr-xr-x   1 19056    wsstaff   143692 Apr  5 18:29 httpd_1.5.1-hpux9.0.5.Z
-rw-r--r--   1 19056    wsstaff   561959 Apr  5 18:29 httpd_1.5.1-hpux9.0.5.tar.Z
-rw-r--r--   1 19056    wsstaff   312943 Apr  5 18:29 httpd_1.5.1-irix4.0.5.tar.Z
-rwxr-xr-x   1 19056    wsstaff   180338 Apr  5 18:29 httpd_1.5.1-irix5.3.Z
-rw-r--r--   1 19056    wsstaff   629811 Apr  5 18:29 httpd_1.5.1-irix5.3.tar.Z
-rwxr-xr-x   1 19056    wsstaff   113227 Apr  5 18:29 httpd_1.5.1-linux1.2.13_ELF.Z
-rw-r--r--   1 19056    wsstaff   523506 Apr  5 18:29 httpd_1.5.1-linux1.2.13_ELF.tar.Z
-rw-r--r--   1 19056    wsstaff   682197 Apr  5 18:29 httpd_1.5.1-osf3.0.tar.Z
-rwxr-xr-x   1 19056    wsstaff   132989 Apr  5 18:29 httpd_1.5.1-solaris2.3_sparc.Z
-rw-r--r--   1 19056    wsstaff   559384 Apr  5 18:29 httpd_1.5.1-solaris2.3_sparc.tar.Z
-rwxr-xr-x   1 19056    wsstaff   121073 Apr  5 18:29 httpd_1.5.1-solaris2.4_sparc.Z
-rw-r--r--   1 19056    wsstaff   538814 Apr  5 18:29 httpd_1.5.1-solaris2.4_sparc.tar.Z
-rwxr-xr-x   1 19056    wsstaff   120421 Apr  5 18:29 httpd_1.5.1-solaris2.4_x86.Z
-rw-r--r--   1 19056    wsstaff   540018 Apr  5 18:29 httpd_1.5.1-solaris2.4_x86.tar.Z
-rwxr-xr-x   1 19056    wsstaff   278669 Apr  5 18:29 httpd_1.5.1-sunos4.1.3.Z
-rw-r--r--   1 19056    wsstaff   725051 Apr  5 18:29 httpd_1.5.1-sunos4.1.3.tar.Z
-rwxr-xr-x   1 19056    wsstaff   258201 Apr  5 18:29 httpd_1.5.1-ultrix4.3.Z
-rw-r--r--   1 19056    wsstaff  1121785 Apr  5 18:30 httpd_1.5.1-ultrix4.3.tar.Z
226 Transfer complete.
ftp> bin bin es para que la transferencia se haga en modo binario
```

```
200 Type set to I.
ftp> get httpd_1.5.1-irix5.3.tar.Z  get es para traer el software
local: httpd_1.5.1-export_irix5.3.Z remote: httpd_1.5.1-
export_irix5.3.Z
200 PORT command successful.
150 Opening BINARY mode data connection for httpd_1.5.1-
export_irix5.3.Z (180338 bytes).
226 Transfer complete.
180338 bytes received in 4.67 seconds (37.75 Kbytes/s)
ftp> 221 Goodbye.
chajul 8%
```

Es importante mencionar que el software que se tiene que bajar es compilado por NCSA y que lo único que tenemos que realizar es la configuración del mismo.

C.5 OBTENIENDO EL SADNS

Para obtener el software hay que accesar al **ftp anónimo** de SOLAR al servidor ftp.sar.net bajo el subdirectorio /pub/sadns y el software es sadns.tar.Z

```
chajul 18% ftp ftp.sar.net
Connected to solar.sar.net.
220 solar FTP server (UNIX(r) System V Release 4.0) ready.
Name (ftp.sar.net:armando): anonymous
331 Guest login ok, send ident as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> cd /pub/sadns
250 CWD command successful.
ftp> get sadns.tar.Z
local: sadns.tar.Z remote: sadns:tar.Z
200 PORT command successful.
150 ASCII data connection for sadns.tar.Z (200.13.64.1,10569)
(983482bytes).
226 ASCII Transfer complete.
ftp>bye
221 Goodbye.
chajul 19%
```

C. 6 CONFIGURANDO EL SADNS

Para configurar el SADNS es necesario configurar el servidor de HTTPD e instalar el software de SADNS en el área del usuario.

Realice la descompresión de los archivos obtenidos de la red.

```
chajul%uncompress *.Z
chajul%
```

5. Se extraen los archivos del archivo del SADNS obtenido (Ver obteniendo el SADNS) . Con el siguiente comando:

```
chajul% uncompress sadns.tar.Z
chajul%
chajul%  tar -xvf sadns.tar
chajul%
```

6. Una vez que se han realizado los pasos anteriores se ejecuta el siguiente comando desde el subdirectorio */usr/people/sadns/wwwsadns* :

```
chajul% httpd
```

O bien desde root para que tome todos los cambios de los procedimientos de mantenimientos.

```
chajul# /user/people/sadns/wwwsadns/httpd
```

7. Finalmente el SADNS está instalado.

C.7 CONFIGURANDO EL ARCHIVO DE ARRANQUE DEL DNS

1. Si se cuenta con los archivos de inicialización del DNS */etc/named.boot* y las demás bases de datos que se encuentran en el archivo *named.boot* (bajo la directiva *directory*), lo que se recomienda es sacar un respaldo de toda la configuración antes de utilizar el SADNS.

2. En caso de no tener la configuración del DNS, desde cualquier cliente de Web (que soporte HTML 3.0), el administrador puede accesar a la página del servidor definido en Configurando SADNS. Inicialmente aparecerá una ventana donde se solicitará el *username* y el *password*. Los datos preconfigurados del sistema es para el acceso del Administrador Local (*username*: root, *password*: sadns01). Una vez que ha ingresado, en la parte de hipertexto de "Herramientas del DNS" elija la opción de "Crea tu propio Servidor de Nombres". Se puede solicitar la ayuda de la pantalla o bien, llenar los datos que nos pide y obtener los archivos de inicialización (Ver Anexo A) y desde la clave de ROOT del sistema UNIX se procede a instalar cada uno de los archivos de configuración en los lugares sugeridos.

3. Después de realizar estos pasos de configuración e instalación, el sistema SADNS está listo para ser usado.

1. Realice la extracción de archivos con el comando tar del servidor de httpd.

```
chajul% tar -xvf httpd_1.5-irix5.3.tar
```

2. Borre el archivo compactado.

```
chajul 35% rm httpd_1.5-irix5.3.tar
chajul 36%
```

3. Cambie de nombre el subdirectorio generado con el tar por **wwwsadns**

```
chajul% mv httpd_1.5.1-export wwwsadns
chajul 39%
```

4. En el subdirectorio ~/wwwsadns/conf los cambios que uno tiene que realizar en los archivos del servidor de Web son en los siguientes: httpd.conf, srm.conf.

   En **httpd.conf** los siguientes datos son importantes:

```
Port 80 (El deseado, el default es 80)
-----
User root
Group #-0
#El User tiene que ser root (capacidad de manipular el DNS)
#El grupo tiene que se el de root
-----
ServerName nuevo.nombre.host (www.sadns.mx en nuestro caso)
-----
ServerAdmin sadns@sar.net
-----
ServerRoot /usr/people/sadns/wwwsadns
-----
#Se comentan las siguientes líneas:

#<VirtualHost 127.0.0.1 Optional>
#DocumentRoot /local
#ServerName localhost.ncsa.uiuc.edu
#ResourceConfig conf/localhost_srm.conf
#</VirtualHost>
```

   En srm.conf se modifican los siguientes datos:

```
DocumentRoot /usr/people/sadns/wwwsadns/bin/docs
----
ScriptAlias /bin/ /usr/local/sadns/wwwsadns/bin/
----
AccessFileName .accesstable
----
```